Minimising Contrastive Divergence with Dynamic Current Mirrors

Chih-Cheng Lu and H. Chen

The Dept. of Electrical Engineering,
The National Tsing-Hua University, Hsin-Chu, Taiwan 30013
hchen@ee.nthu.edu.tw

Abstract. Implementing probabilistic models in Very-Large-Scale-Integration (VLSI) has been attractive to implantable biomedical devices for improving sensor fusion. However, hardware non-idealities can introduce training errors, hindering optimal modelling through on-chip adaptation. This paper investigates the feasibility of using the dynamic current mirrors to implement a simple and precise training circuit. The precision required for training the Continuous Restricted Boltzmann Machine (CRBM) is first identified. A training circuit based on accumulators formed by dynamic current mirrors is then proposed. By measuring the accumulators in VLSI, the feasibility of training the CRBM on chip according to its minimizing-contrastive-divergence rule is concluded.

Keywords: Minimising Contrastive Divergence, Dynamic Current Mirrors, Probabilistic Model, Boltzmann Machine, On-chip training.

1 Introduction

As probabilistic models are able to generalise the natural variability in data, the VLSI implementation of probabilistic models has been attractive to implantable biomedical devices [1] [2]. However, seldom probabilistic models are amenable to the VLSI implementation. Among the proposed probabilistic models in VLSI [3] [4] [5], the Continuous Restricted Boltzmann Machine (CRBM) has been shown capable of modelling biomedical data with a hardware-friendly training algorithm, which minimises the contrastive divergence (MCD) between training and modelled distributions [6] [7]. However, experiments in [7] revealed that offsets in training circuits limited the minimum achievable divergence, preventing the CRBM microsystem from modelling data optimally. To make useful the VLSI implementation of the CRBM and many other models, it is important to develop a simple circuit capable of realising contrastive training rules with satisfactory precision.

This paper examines the feasibility of using dynamic current mirrors to realise contrastive-divergence training algorithms on-chip with satisfactory precision. As continuous-valued models are inherently more sensitive to the existence of training errors, the satisfactory precision refers to the capability of training the CRBM microsystem to model both artificial and real biomedical (ECG) data satisfactorily.

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 410–420, 2009.

2 The CRBM Model

The CRBM consists of one visible and one hidden layers of stochastic neurons with inter-layer connections only [7]. The number of visible neurons corresponds to the dimension of data, while that of hidden neurons is chosen according to data complexity [7]. Let w_{ij} represent the bi-directional connection between neurons s_i and s_j. The stochastic state of a neuron s_i is defined by [7]

$$s_i = \varphi_i(a_i \cdot (\Sigma_j w_{ij} \cdot s_j + N_i(0,\sigma))) \tag{1}$$

where $N_i(0,\sigma)$ represents a Gaussian noise with zero mean and variance σ^2, and $\varphi_i(\cdot)$ a sigmoid function (e.g. $tanh(\cdot)$) with asymptotes at ± 1. Parameter ai controls the slope of the sigmoid function and thus the variance of s_i, such that the neuron is either near-deterministic (small a_i), or continuous-stochastic (moderate a_i), or binary-stochastic (large a_i). Let λ represent the parameter $\{w_{ij}\}$ or $\{a_i\}$. Parameters in a CRBM microsystem are trained by the simplified MCD algorithm [3]

$$\triangle\lambda = \eta_\lambda \cdot (\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4) \tag{2}$$

where $\hat{s_i}$ and $\hat{s_j}$ denotes the one-step Gibbs-sampled states [7], η_λ the updating rate, and $\langle\cdot\rangle_4$ taking the expectation over four training data. The difference between $\langle s_i \cdot s_j \rangle_4$ and $\langle \hat{s_i} \cdot \hat{s_j} \rangle_4$ corresponds to the contrastive divergence between training and modelled distributions [6] and has to be minimised. For training $\{a_i\}$, s_j and $\hat{s_j}$ in Eq.(2) are replaced by s_i and $\hat{s_i}$, respectively.

3 Maximum Offsets Tolerable by the CRBM

The CRBM has been realised as a VLSI microsystem containing six neurons with on-chip training circuits [3]. However, hardware nonidealities in training circuits prevents the CRBM system from modelling data optimally, and it was shown that the overall effect of hardware nonidealities can be modelled as the "biased" training algorithm

$$\triangle\lambda = \eta_\lambda \cdot (\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4 + \triangle_T) \tag{3}$$

where $\triangle_T$ represents the offset that limits the minimum contrastive divergence achievable by on-chip training circuits. Although the offset varies from one circuit to another, it is assumed to be identical in simulation for simplicity. Based on the software-hardware mapping derived in [3], the following subsections simulate the behaviour of a CRBM microsystem with Eq.(3), and identify the maximum offsets ($\triangle_T$) the system can tolerate. The value of tolerable offsets will be given in terms of percentage, normalized with respect to the maximum value of $|\langle s_i \cdot s_j \rangle_4| = 1$. (i.e. $\triangle_T = 1$ % refers to $\triangle_T = 0.01$ in Eq.(3)).

3.1 Quantitative Index for Offset Tolerance

As a *generative* model, the CRBM learns to "regenerate" training data distribution.To identify an index for measuring quantitatively how well the CRBM models a dataset, the

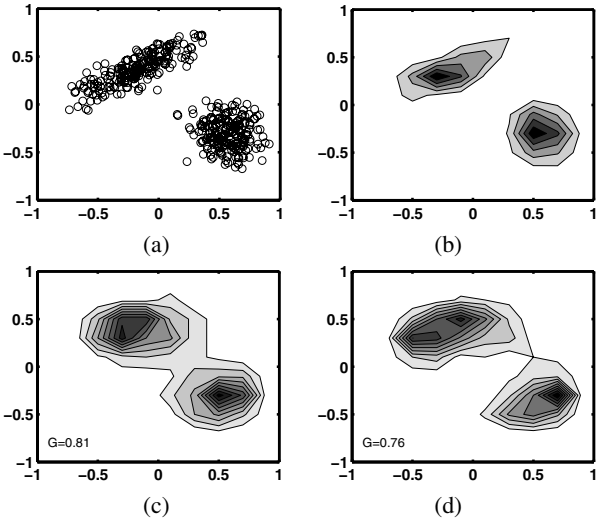

Fig. 1. (a) Artificial training data with two clusters of Gaussian-distributed data points. (b) The statistical density of the training data. (c)(d): The statistical density of 20-step reconstructions generated by the CRBM after (c)20000 (d)30000 training epochs.

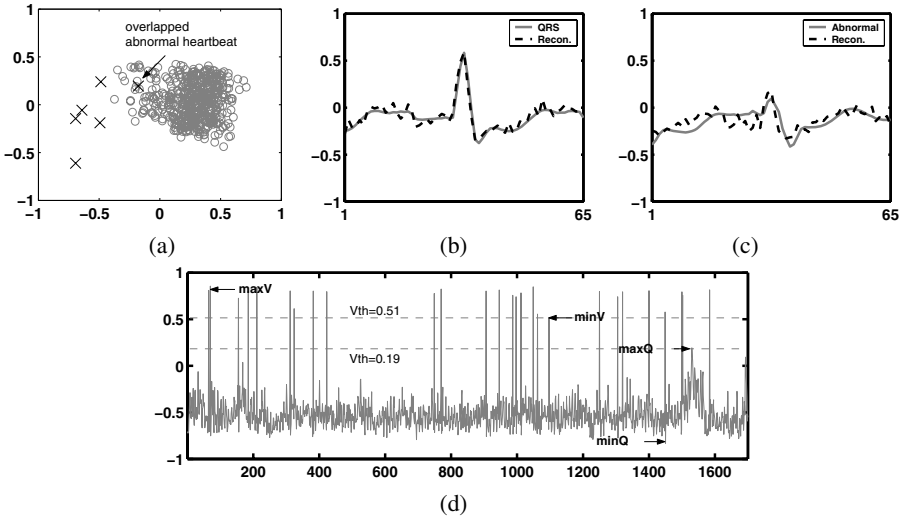

Fig. 2. (a) The projection of 500 ECG training data to its first two principle components. The projection of the five abnormal ECGs are denoted by black crosses. (b)(c)(d): Results of training the CRBM to model ECG data with $\triangle_T = 0.2\%$ for all parameters. (b) (b) The normal and (c)the abnormal ECGs in training dataset (grey) and the reconstruction by the trained CRBM (dashed). (d) Responses of hidden neuron h_3 to 1700 testing data. maxV, minV, maxQ, and minQ correspond to the maximum and minimum responses to abnormal heartbeats, and maximum and minimum responses to normal heartbeats, respectively.

Table 1. Tolerable offsets with four-data, sign-valued training algorithm, four-data, real-valued training algorithm, and single-datumn training algorithm

METHOD\MODELLED DATA	TWO-CLUSTER	ECG
Four-data, sign-valued	1%	0.2%
Four-data, real-valued	1%	0.3%
Single-data, real-valued	0.2%	0.05%

CRBM with two visible and four hidden neurons, as shown in Fig.1, was first trained with the ideal algorithm (Eq.(2)) to model the artificial data in Fig.1(a). The dataset contains one elliptic and one circular clusters of 200 Gaussian-distributed data points whose statistical density is shown in Fig.1(b). Let $P^T(\mathbf{v})$ and $P^M(\mathbf{v})$ represent the distribution of training data and the distribution modeled by the CRBM, respectively. The *Kulback-Leibler (KL) Divergence* defined as Eq.(4) [8] measures the difference between $P^T(\mathbf{v})$ and $P^M(\mathbf{v})$.

$$G = \Sigma_{\mathbf{v}} P^T(\mathbf{v}) log \frac{P^T(\mathbf{v})}{P^M(\mathbf{v})} \tag{4}$$

where $\mathbf{v}$ denotes the subset of visible states, and G equals zero when $P^T(\mathbf{v}) = P^M(\mathbf{v})$. As not all distributions can be described by explicit equations, $P^T(\mathbf{v})$ and $P^M(\mathbf{v})$ were statistically-estimated by dividing the two-dimensional space into 10x10 square grids, counting the number of data points in each grid, and normalising the counts with respect to the total number of data points. Fig.1(c)(d) shows the statistical density of 20-step reconstructions generated by the CRBM after 20000 and 30000 training epochs, respectively. The G values calculated according to Eq.(4) are shown at the bottom-left corner of each subfigure, indicating that the KL-divergence is a reliable index for measuring quantitatively the similarity between training and modelled distributions. Similar results are obtained for other data like doughnut-shaped distribution. As the training updates of most parameters become negligible after $G < 0.8$, it is chosen as the criterion for identifying the tolerable offsets for the CRBM. When all parameters ($\{w_{ij}\}$, $\{a_{vi}\}$, and $\{a_{hi}\}$) experience offsets, the maximum offsets the CRBM can tolerate to model artificial data was identified to be 1% (Table 1).

3.2 Modelling Real Heartbeat Data with Offsets

The tolerable offset for modelling high-dimensional, real-world data was examined in the context of recognising electrocardiograms (ECG), extracted from the MIT-BIH database as in [9] [10]. The training dataset contains 500 heartbeats with only 5 abnormal heartbeats. The testing dataset contains 1700 heartbeats with 27 abnormal heartbeats. Each ECG trace was sampled as a 65-dimensional datum, and Fig.2(a) shows the projection of the training dataset onto its first two principle components. Although the dimension reduction made the quantitative index G remain applicable, pilot simulation showed that modelling training data satisfactorily did not guarantee the detection of abnormal heartbeats with 100% accuracy. This was because the projected distributions of normal and abnormal heartbeats at low dimension overlap with each other, as obviated

by Fig.2(a). (the data remain separable in high dimensions). Therefore, detection with 100% accuracy was used as a stricter criterion for identifying the tolerable offsets for modelling ECG data. Extensive simulations further showed that the CRBM tolerated an offset of only 0.2% to model ECG data (Table 1). With $\triangle_T = 0.2\%$, the trained CRBM was able to reconstruct both normal and abnormal ECG signal satisfactorily, as shown in Fig.2(b)(c). In addition, Fig.2(d) shows the responses of hidden neuron h_3 to 1700 testing data $\{\mathbf{d}\}$, calculated according to Eq.(5). The abnormal heartbeats can be detected with 100% accuracy by setting any threshold between minV and maxQ.

$$h_3 = \varphi(a_3 \cdot (\mathbf{w}^{(3)} \cdot \mathbf{d})) \tag{5}$$

3.3 Tolerable Offsets for Different Training Strategies

As implementing training circuits with an offset less than 0.2% is quite a challenge, we further investigated the possibility of releasing the strict requirement with two modified training strategies, (a)updating parameters with real-valued contrastive divergence $(\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4)$ instead of taking only its sign and (b)updating parameters with real-valued contrastive divergence datum by datum [11]. The last two rows of Table 1 summarise the maximum offsets the CRBM can tolerate to model artificial and ECG data with the different training strategies. Comparison with the first row indicates that four-data, real-valued adaptation does enhance the tolerance against offsets slightly, while single-datum adaptation degrades the tolerance significantly. The latter demonstrates that calculating the expectation value, i.e. accumulating opinions from multiple data, is important for estimating the "contrastive divergence" between distributions.

4 The Contrastive-Divergence Training Circuit Based on Dynamic Current Mirrors

Although an offset smaller than 0.3% remains challenging, the dynamic current mirrors (DCMs) were reported to have errors smaller than 500ppm [12] [13]. Therefore, we propose the training circuit in Fig.3 that uses DCMs to calculate $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4$ or $(s_i \cdot s_j - \hat{s_i} \cdot \hat{s_j})$ in contrastive-divergence training rules. Each DCM works as a register, using the same transistor (M1 or M2 in Fig.3(a)) to sample and transfer currents. The mismatching errors in conventional current mirrors, i.e. the main cause of training offsets in [10], are thus avoided. Fig.3(a) shows the accumulator consisting of one NMOS (M1-M1c) and one PMOS (M2-M2c) DCMs. The DCM training circuit realises the single-datum training algorithm by simply three steps. I_{in} in Iin in Fig.3(a) represents $s_i \cdot s_j$ or $\hat{s_i} \cdot \hat{s_j}$ calculated by the multiplier. At the first step with switches S_{IN}, S_{D1}, and S_{G1} cloased, the current representing $s_i \cdot s_j$ is sampled into the NMOS DCM, and then stored as the voltage across the capacitor $C1$ after S_{IN} and S_{G1} become opened. At the second step, switches S_{D2} and $SG2$ are closed to copy the same current into the PMOS DCM. The sampled current is stored as a voltage across the capacitor $C2$ after S_{D2} and S_{G2} are opened. At the third step, I_{in} representing $\hat{s_i} \cdot \hat{s_j}$ is sampled and stored in the NMOS DCM by repeating the first step. Finally, closing S_{D1}, S_{D2}, and S_{OUT} gives an output current proportional to $(s_i \cdot sj - \hat{s_i} \cdot \hat{s_j})$. To implement the four-data

training algorithm, the circuit in Fig.3(a) functions as an accumulator that calculates $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s}_i \cdot \hat{s}_j \rangle_4$. The first and the second steps described above are carried out once to store $s_i \cdot s_j$ of the first datum into the PMOS DCM. At the third step, let I_{in} correspond to $s_i \cdot s_j$ of the second datum. Closing S_{IN}, S_{D1}, S_{G1}, and S_{D2} stores the sum of $s_i \cdot s_j$ of both data into the NMOS DCM. Repeating the second and the third steps alternatively then sums up $s_i \cdot s_j$ of multiple data and stores it into the NMOS DCM. To calculate the contrastive divergence, $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s}_i \cdot \hat{s}_j \rangle_4$ two identical accumulators are employed as in Fig.4(b). The top accumulator calculates $\langle s_i \cdot s_j \rangle_4$ and stores the value into its PMOS DCM, while the bottom one simply stores $\langle \hat{s}_i \cdot \hat{s}_j \rangle_4$ into NMOS DCM. As soon as switch SOUT is closed, an output current proportional to $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s}_i \cdot \hat{s}_j \rangle_4$ is produced. Finally, I_{OUT} is directed into the charge amplifier in Fig.3(c) which functions as the low-impedance reference voltage (V_{REF}) in Fig.3(b). I_{OUT} hen modifies the voltage stored across C_F , which represents a parameter value of the CRBM.

The DCM training circuit in Fig.3 is capable of realising the three contrastive-divergence training algorithms in Table 1, as well as other contrastive training rules in [8] [14] [15] [16]. The learning rate η_λ in Eq.(2) can be defined by the period of closing S_{OUT}. Unlike the DCMs in [12] [13], the DCMs in Fig.3 have not only to transport currents of various values but also to function as both accumulators and subtractors, coping with a wide range of currents. Cascode transistors M1a and M2a in Fig.3(a) are therefore employed to reduce the effect of channel-length modulation by fixing the drain voltage of M1 and M2, respectively.

The DCM training circuit also suffers from offsets introduced by the nonlinearity of multipliers and the charge-injection errors. The former can be easily avoided by using a multiplier with symmetric outputs, for example, the modified Chible multiplier proposed in [3]. As for the latter, complementary transistors can be used as switches to compensate for charge-injection errors. However, the simulation normally underestimates the charge-injection errors. The DCM accumulators in Fig.3(a) and (b), excluding the multiplier, were thus fabricated with the TSMC 0.35um 2P4M CMOS process to investigate the precision achievable by the proposed training circuits.

For the single-datum training algorithm, I_{in} was designed to range from $1\mu A$ to $3\mu A$ corresponding to $s_i \cdot s_j = -1$ and $\hat{s}_i \cdot \hat{s}_j = 1$, respectively. To minimise the dependence of charge injection on I_{in}, the charge-injection error is minimised at $I_{in} = 2\mu A$. For four-data training algorithm, I_{in} was designed to range from $0.25\mu A$ to $0.75\mu A$, such that the accumulation of four data still ranges from $1\mu A$ to $3\mu A$, allowing the minimisation of charge-injection error to remain at $I_{in} = 2\mu A$.

5 Measurement Results

With I_{in} generated from a Source Meter (Keithley 2602), the output of the DCM accumulators were connected to a current-voltage(I-V) converter, which emulated V_{REF} in Fig.3 and converted I_{out} into a voltage of $V_o = V_{REF} - H_f \cdot I_{out}$ with $H_f = 1650(V/A)$. The voltage change $\triangle V_o = V_o - V_{REF} = -H_f \cdot I_{out}$ at the instant of closing S_{OUT} was then measured. With digital-control clocks generated by a Field-Programmable-Gate-Array (FPGA) chip, the DCM accumulators were set easily to calculate $(s_i \cdot sj - \hat{s}_i \cdot \hat{s}_j)$ or $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s}_i \cdot \hat{s}_j \rangle_4$.

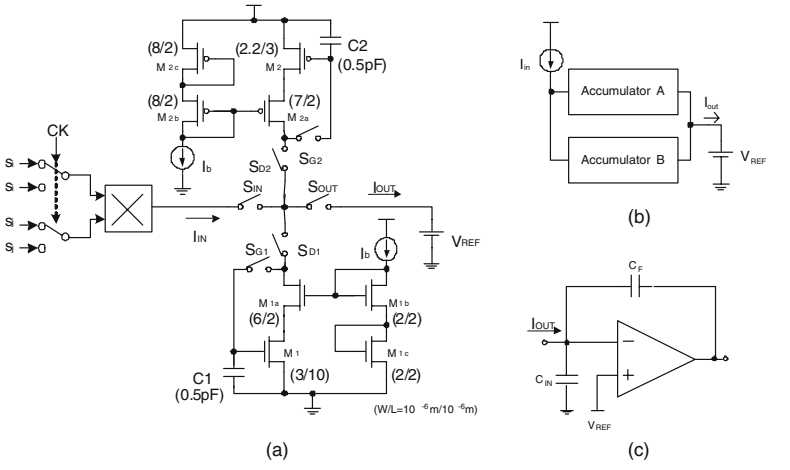

Fig. 3. The proposed DCM training circuit consisting of (a) a multiplier and the DCM accumulator calculating $(s_i \cdot sj - \hat{s_i} \cdot \hat{s_j})$, or (b) the DCM accumulator calculating $(\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4)$, and, (c) the charge amplifier

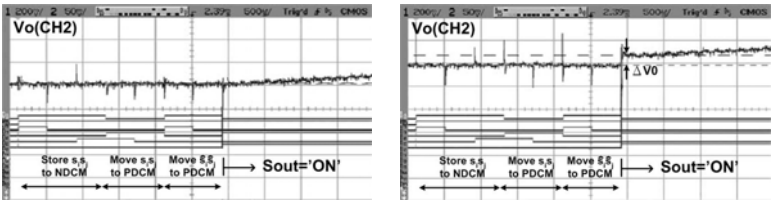

Fig. 4. Measured $V_0 = V_{REF} - I_{out} \cdot H_f$ for single-datum training with $I_{in} = 1\mu A$. (a) without offset (b) with offset. The digital signals from top to bottom correspond to S_{IN}, S_{D1}, S_{G1}, S_{D2}, S_{G2}, and S_{OUT} in Fig.3.

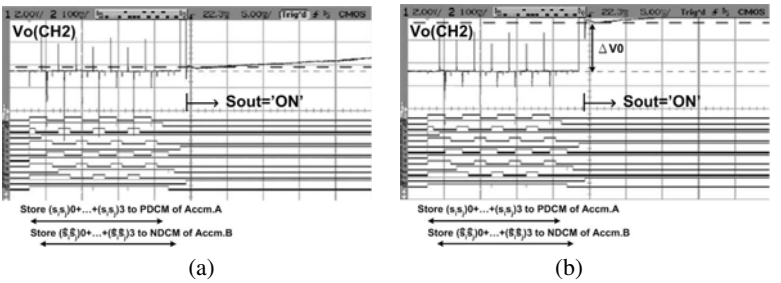

Fig. 5. Measured $V_0 = V_{REF} - I_{out} \cdot H_f$ for four-data training with $I_{in} = 0.25\mu A$. (a) without offset (b) with offset. The digital signals from top to bottom correspond to S_{IN}, S_{D1}, S_{G1}, S_{D2}, S_{G2}, and S_{OUT} for accumulator A and those of accumulator B in Fig.3.

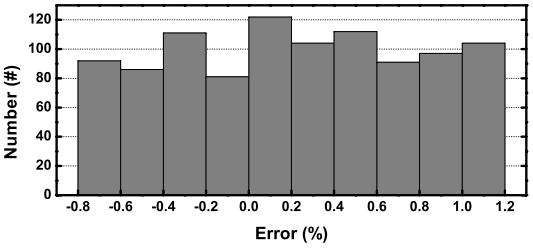

Fig. 6. Statistical histogram of the offset errors measured from a DCM training circuit set to carry out single-data training algorithm

Table 2. TMeasured offsets in the calculation of $(s_i \cdot sj - \hat{s_i} \cdot \hat{s_j})$

$s_i \cdot s_j$	$\hat{s_i} \cdot \hat{s_j}$	MEAN ERROR	STD. DEV.
$1\mu A$	$1\mu A$	0.38%	0.56%
$2\mu A$	$2\mu A$	0.19%	0.39%
$3\mu A$	$3\mu A$	0.45%	0.63%

Table 3. Measured offsets in the calculation of $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4$

$s_i \cdot s_j(\mu A)$	$\hat{s_i} \cdot \hat{s_j}(\mu A)$	MEAN ERROR	STD. DEV.
0.25+0.25+0.25+0.25	0.25+0.25+0.25+0.25	1.31%	0.38%
0.5+0.5+0.5+0.5	0.5+0.5+0.5+0.5	2.21%	0.55%
0.75+0.75+0.75+0.75	0.75+0.75+0.75+0.75	3.61%	0.63%

For calculating $(s_i \cdot sj - \hat{s_i} \cdot \hat{s_j})$, the measured V_o in response to a constant I_{in} of $1\mu A$, i.e. $s_i \cdot s_j = \hat{s_i} \cdot s_j = -1$, is shown in Fig.4. Although $\triangle V_0$ ideally equaled zero, $\triangle V_0$ measured from the same circuit either approximated zero or varied from one trial to another. Similar results were observed when calculating $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4$, as shown in Fig.5. Normalising $\triangle V_0$ with respect to $1.65V$ (the $\triangle V_0$ for $I_{out} = 1\mu A$ representing $|S_i \cdot s_j| = 1$) gives the offset errors in terms of percentage. Fig.6 shows the statistical distribution of the offsets measured from 1000 trials of calculating $(s_i \cdot s_j - \hat{s_i} \cdot \hat{s_j})$. Interestingly, the offsets exhibit a uniform distribution instead of staying constant, and the variance is greater than the mean value.

The offsets in the DCM accumulators were caused by charge-injection errors, leakage currents of C1 and C2, and clock jitters generated by the FPGA chip. To investigate the contribution of leakage currents, two types of digital clocks were used to buffer I_{in} to I_{out} by storing I_{in} in the NMOS DCM, transferring it to the PMOS DCM, and subsequently outputing the current. One clock differs from the other mainly by shortening the period of opening S_{G1} and S_{G2}. Shortening the period improved the mean errors from -8.09% to -6.38%, while the standard deviations of the two cases are comparable

(3.45%). Therefore, leakage currents mainly affect the mean errors, while clock jitters have dominant effects on the variance. Accumulating four I_{in} caused mean errors to increase by more than four times, while the standard deviations remained about the same. Charge injection thus also affected mainly the mean errors.

Table 2 summarises the performance of the DCM accumulator in calculating $(s_i \cdot sj - \hat{s_i} \cdot \hat{s_j})$ with different I_{in}. The mean errors became significantly smaller than 6.38%, indicating that charge-injection and leakage-current errors in the NMOS and PMOS DCMs cancelled with each other largely through the subtraction operation. Complete cancellation was difficult because the current representing $(s_i \cdot sj$ in the PMOS DCM unavoidably suffered from extra switching events than the current representing $\hat{s_i} \cdot \hat{s_j})$ in the NMOS DCM. Moreover, Table 2 reveals that the randomness in offsets was also reduced by the subtraction operation. Table 3 summarises the performance in calculating $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4$. Compared to Table 2, the mean errors all became higher because charge-injection and leakage-current errors were accumulated. Nevertheless, the standard deviations in both tables are comparable, confirming that clock jitters dominate to introduce the randomness in offsets.

6 Modelling Data with Uniformly-Distributed Offsets

To simulate the performance of a CRBM microsystem with the proposed DCM training circuits, $\triangle_T$ in the training rule was replaced by a uniform random variable with nonzero mean (μ_T) and a standard deviation (σ) of 0.7%, the measured maximum deviation. Under the existence of $\triangle_T$ for all parameters, the maximum mean offsets (μ_T) that the CRBM can tolerate to model both artificial and ECG data are identified and summarised in Table 4. Compared to Table 1, the tolerance is slightly improved. This feature agrees with the finding that randomness releases the precision required for training a multi-layer-perceptron [17] [18]. The CRBM is able to correct training errors whenever the random offset is small, and thus to discourage the saturation of parameter values. Therefore, it is important to know when to stop training once the data distribution is modelled, so as to prevent $\triangle_T$ from dominating to causes all parameters to saturate. Fortunately, the G value could be used as a reliable indicator for when to stop.

Table 4. Tolerable mean offsets by the CRBM with different training strategies

METHOD\MODELLED DATA	TWO-CLUSTER	ECG
Four-data, sign-valued	2%	0.3%
Four-data, real-valued	2%	0.5%
Single-data, real-valued	0.5%	0.1%

7 Conclusion

The feasibility of minimising contrastive divergence on-chip with DCMs has been carefully investigated by both behavioural simulation of the CRBM microsystem and the

VLSI implementation of DCM accumulators. The simulation indicates that the CRBM can tolerate a maximum offset of only 0.3% to model real biomedical (ECG) data satisfactorily, and that the tolerance can be slightly-improved by real-valued adaptation. On the other hand, measurement results of DCM accumulators indicate that the accumulation errors in DCMs can be largely cancelled by the subtraction operation essential for the contrastive-divergence training. As the mean offsets in Table II are all smaller than 0.5%, i.e. the tolerable offset for four-data, real-valued trainning in Table 4, using four DCM accumulators (Fig.3(a)) to calculate $\langle s_i \cdot s_j \rangle_4 - \langle \hat{s_i} \cdot \hat{s_j} \rangle_4$ would allow us to avoid the accumulation error in Table 3 while achieving satisfactory precision. This suggestion will be further confirmed with the VLSI implementation of the full training circuit.

Acknowledgement

The authors would like to acknowledge the TSMC and CIC for fabrication of the chip, and the National Science Council (NSC) in Taiwan for funding this project (Grant code: NSC 95-2221-E-007-115).

References

1. Schwartz, A.B.: Cortical Neural Prothetics. Annual Review Neuroscience, 487–507 (2004)
2. Lebedev, M.A., Nicolelis, M.A.L.: Brain-machine interfaces: past, present and future. TRENDS in Neuroscience 29[9], 536–546 (2006)
3. Chen, H., Fleury, P., Murray: Continuous-Valued Probabilistic Behaviour in a VLSI Generative Model. IEEE Trans. on Neural Networks 17(3), 755–770 (2006)
4. Genov, R., Cauwenberghs, G.: Kerneltron: support vector machine in silicon. IEEE Trans. on Neural Networks 14(8), 1426–1433 (2003)
5. Hsu, D., Bridges, S., Figueroa, M., Diorio, C.: Adaptive Quantization and Density Estimation in Silicon. In: Advances in Neural Information Processing Systems (2002)
6. Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation 14(8), 1771–1800 (2002)
7. Chen, H., Murray, A.F.: A Continuous Restricted Boltzmann Machine with an Implementable Training Algorithm. IEE Proc. of Vision, Image and Signal Processing 150(3), 153–158 (2003)
8. Hinton, G.E., Sejnowski, T.J.: Learning and Relearning in Boltzmann Machine. In: Rumelhart, D., McClelland, J.L., The PDP Research Group (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, pp. 283–317. MIT, Cambridge (1986)
9. MIT-BIH Database Distribution, `http://ecg.mit.edu/index.htm`
10. Chen, H., Fleury, P., Murray, A.F.: Minimizing Contrastive Divergence in Noisy, Mixed-mode VLSI Neurons. In: Advances in Neural Information Processing Systems, vol. 16 (2004)
11. Chiang, P.C., Chen, H.: Training Probabilistic VLSI models On-chip to Recognise Biomedical Signals under Hardware Nonidealities. In: IEEE International Conf. of Engineering in Medicine and Biology Society (2006)
12. Wegmann, G., Vittoz, E.: Analysis and Improvements of Accurate Dynamic Current Mirrors. IEEE J. of Solid-State Circuits 25[3], 699–706 (1990)
13. Fleury, P., Chen, H., Murray, A.F.: On-chip Contrastive Divergence Learning in Analogue VLSI. In: Proc. of the International Joint Conference on Neural Networks (2004)

14. Teh, Y.W., Hinton, G.E.: Rate-coded Restricted Boltzmann Machine for Face Recognition. In: Advances in Neural Information Processing System. MIT Press, Cambridge (2001)
15. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Englewood Cliffs (1998)
16. Peterson, C., Anderson, J.R.: A Mean Field Theory Learning Algorithm for Neural Networks. Complex Systems 1, 995–1019 (1987)
17. Murray, A.F.: Analogue Noise-enhanced Learning in Neural Network Circuits. Electronics Letters 27(17), 1546–1548 (1991)
18. Murray, A.F., Edwards, P.J.: Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training. IEEE Trans. on Neural Networks 5(5), 792–802 (1994)