CDMTCS Research Report Series Supplemental Abstracts for The 3nd International Conference on Unconventional Models of Computation, UMC'02

C. S. Calude and M. J. Dinneen
Department of Computer Science,
University of Auckland,
Auckland, New Zealand

F. Peper
Communications Research Laboratory,
Nanotechnology Group,
Kobe, Japan

(Editors)

CDMTCS-195
October 2002
Centre for Discrete Mathematics and
Theoretical Computer Science

Editors' Notes

These are additional abstracts of talks that were presented at the *Third International Conference on Unconventional Models of Computation (UMC'O2).* UMC'02, organized by the Centre for Discrete Mathematics and Theoretical Computer Science, and the Kansai Advanced Research Center of the Communications Research Laboratory was held at the Orbis hall in the center of Kobe's Rokko island, during October 15–19, 2002.

The conference encompasses all areas of unconventional computation, especially quantum computing, DNA-based computation, evolutionary algorithms and other proposals for computation models that go beyond the Turing model. Additional refereed and invited papers of UMC'02 appear in the following proceedings:

> C. S. Calude, M. J. Dinneen, and F. Peper editors. *Unconventional Models of Computation, UMC'02*, LNCS 2509, Springer-Verlag, October 2002, VIII + 300 pages.

Qubit Neural Networks

Noriaki Kouda[1], Nobuyuki Matsui[1], Haruhiko Nishimura[2]
, and Ferdinand Peper[3]

[1] Department of Computer Engineering, Graduate School of Engineering,
Himeji Institute of Technology,
2167 Shosha, Himeji, Hyogo, 671-2201,Japan
[2] Studies of Information Science, Hyogo University of Education,
942-1 Shimokume, Yashiro, Hyogo, 673-1494 Japan
[3] Communications Research Laboratory, Ministry of Posts and Telecommunications,
Auditory and Visual Informatics Group,
588-2, Iwaoka, Iwaoka-cho, Nishi-ku, Kobe, Hyogo, 651-2492, Japan

Though neural networks have attracted much interest in the last two decades for their potential to describe brain function realistically, they have failed thus far to provide models that can be simulated in a reasonable time on computers, other than toy models. Quantum computing is a likely candidate for improving the computational efficiency of neural networks, since it has been very successful in doing so for a selected set of computational problems. In this framework, the Qubit neuron model, proposed by Matsui and Nishimura, has shown its promise in improving the efficiency in problems like data compression. In this paper, we apply it on learning two functional descriptions, the 4-bit parity function and a continuous function, by the so-called Back-Propagation learning method. The simulations we perform show that the Qubit model performs these tasks with significantly improved efficiency as compared to the classical model, and a classical model in which complex numbers are allowed as model parameters and as input (See Fig.1). Our simulations suggest that the improved performance is due to the use of superposition of neural states and the use of the probability interpretation in the observation of the output states of the model.

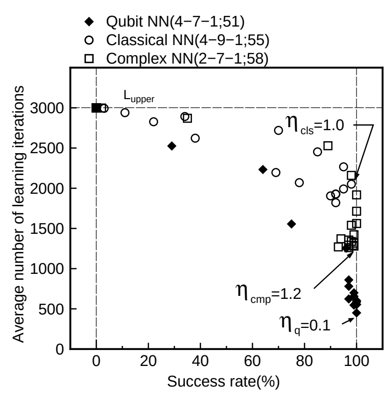

Fig. 1. Learning performance in 4-bit parity check problem. The proposed Qubit Neural Network requires substantially less learning iterations on the average to achieve the same success rates.

Expressive Power of Quantum Pushdown Automata with a Classical Stack

Masaki Nakanishi[1*], Takao Indoh[2],
Kiyoharu Hamaguchi[3], and Toshinobu Kashiwabara[3]
[1]Nara Institute of Science and Technology, Japan
[2]Fujitsu Co. Ltd., Japan
[3]Osaka University, Japan
[*]E-mail: m–naka@is.aist-nara.ac.jp

1 Introduction

As one of quantum computation models, quantum pushdown automata have been proposed by Golovkins [1]. In [1], it has been shown that quantum pushdown automata can recognize every regular language and some non-context-free languages. Their model has a quantum tape head and a quantum stack, and needs $O(n)$ qubits for realization, where n is the execution time. We introduce another model of quantum pushdown automata whose stack is assumed to be implemented as a classical device. This means that our model needs $O(\log m)$ qubits for specifying the position of the head and representing the finite state control, where m is an input length.We show that our quantum pushdown automaton model can recognize some non-context-free languages with arbitrarily large probability. We also show that our model can simulate any (non-reversible) probabilistic pushdown automata with the same probability as that of the original probabilistic pushdown automata.

2 Quantum Pushdown Automata

Definition 1 A Quantum Pushdown Automaton with a Classical Stack (QPA-CS) is defined as the following 8-tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, \sigma, Q_{\rm acc}, Q_{\rm rej})$, where Q is the set of states, Σ is the set of input symbols including the left and the right end-marker $\{¢, \$\}$ respectively, Γ is the set of stack symbols including the bottom symbol Z, δ is the quantum state transition function ($\delta : (Q \times \Sigma \times \Gamma \times Q \times \{0,1\}) \longrightarrow \mathbb{C}$), q_0 is the initial state, σ is the function by which words to be pushed are determined ($\sigma : Q \setminus (Q_{\rm acc} \cup Q_{\rm rej}) \longrightarrow \Gamma^*$), $Q_{\rm acc} (\subseteq Q)$ is the set of accepting states, and $Q_{\rm rej} (\subseteq Q)$ is the set of rejecting states, where $Q_{\rm acc} \cap Q_{\rm rej} = \emptyset$. □

We define the observable $\mathcal{O}$ as $\mathcal{O} = \oplus_j E_j$, where $E_j = span\{|q,k\rangle \mid \sigma(q) = j\}, E_{\rm acc} = span\{|q,k\rangle \mid q \in Q_{\rm acc}\}, E_{\rm rej} = span\{|q,k\rangle \mid q \in Q_{\rm rej}\}$. At every step, the finite state control and the tape head of a QPA-CS evolve according to the state transition function δ, and are observed with respect to the observable $\mathcal{O}$. Then, the stack top symbol is popped and the outcome of the observation is pushed, or it halts outputting 'acc' or 'rej'.

3 On Expressive Power of QPA-CS's

By using the technique shown in [2], we construct QPA-CS's that recognize the non-context-free language $L = \{a^m db^n dc^n | m \geq n \geq 0\}$. It is straightforward to prove that L is not a context-free language by the pumping lemma.

Theorem 1 QPA-CS's can recognize the language L.
(Sketch of Proof) The substring $b^n dc^n$ of $a^m db^n dc^n$ can be accepted with arbitrarily large probability by using the technique in [2], in which it is shown that 2-way quantum finite automata can recognize $\{a^n b^n\}$. Comparison of m and n can be done using the stack as a counter. □

Next, we show that QPA-CS's can simulate any probabilistic pushdown automata. We say a Probabilistic Pushdown Automaton (PPA) to be of post-state-dependent type if the words to be pushed and the direction of the head movement are determined solely by the next state rather than by the triple of (current) state, input symbol and stack top symbol, where the next state is the state after transition. Then we have the following lemma.

Lemma 1 Any PPA can be converted to a PPA of post-state-dependent type.
(Proof) Omitted. □

By using Lemma 1, we have the following theorem.

Theorem 2 A QPA-CS can simulate any PPA with the same probability as that of the original PPA.
(Sketch of Proof) We show that any PPA can be made to be reversible, where *reversible* means that each state has at most one incoming transition for any stack top symbol. We assume that a PPA M is of post-state-dependent type without loss of generality by Lemma 1. We also assume that a state q has multiple incoming transitions that are activated by the same stack top symbol y. Let each of these transitions be $\delta(q', x, y, q, w, d) = p$, where q', x, w

,d and p are the source state, the input symbol, the word to be pushed, the direction of the head movement and the probability associated with the transition respectively. Then we replace this transition with $\delta(q', x, y, q'', s_{q''}, 0) = p$ and $\delta(q'', x, s_{q''}, q, w, d) = 1$, where q'' and $s_{q''}$ are uniquely associated with the original transition. Note that this modification does not change the computation of the PPA M, and makes each state have at most one incoming transition for any stack top symbol. Thus we have a reversible PPA. $\square$

4 Conclusion

We have shown that QPA-CS's can recognize the non-context-free language L with arbitrarily large probability and can simulate any PPA with the same probability as that of the original PPA.

References

[1] M.Golovkins, "Quantum Pushdown Automata," LANL e-print quant-ph/0102054, 2001.

[2] A.Kondacs and J.Watrous, "On the power of quantum finite state automata," Proc. 38th Symp. on Foundations of Computer Science, pp.66–75, 1997.