Neural Network Parallel Simulation with Logic Programming
(*Extended Abstract*)

Priscila M. V. Lima and Inês C. Dutra

Department of Systems Engineering and Computer Science
Federal University of Rio de Janeiro
`{priscila,ines}@cos.ufrj.br`
`http://www.cos.ufrj.br/{priscila,ines}`

Este trabalho apresenta a implementação de uma Rede Neuronal Artificial para o processamento de inferência. O sistema foi escrito na linguagem Parlog [3] e modela três tipos de simulação: baseada no modelo de Hopfield, em máquinas de Boltzman e máquinas de Boltzman com hiper arcos.

Os simuladores implementam redes que podem apresentar qualquer topologia. Cada neurônio é representado por um processo em Parlog (objetivos no corpo de uma cláusula) e a comunicação é feita através da variável lógica (*Stamped Addressed Envelopes* – SAE [18]). De forma simplificada, a arquitetura da rede é tal que cada vez que um neurônio n_i está pronto para operar, além de enviar seu novo estado para o vizinho n_j, também envia um "slot" vazio (SAE) para o retorno do novo estado computado por n_j. Existe um SAE por cada par de neurônios vizinhos.

Grupos de neurônios no simulador podem formar hiper arcos. Neste caso, estes hiper arcos são transformados em cliques. Cada par de neurônios vizinhos continua compartilhando um SAE mesmo que estejam envolvidos em mais de um arco.

A rede pode ser configurada de três formas distintas:

- Modelagem: atribuição de pesos feita de forma manual [1, 8].
- Síntese: através da equação de energia [15, 9].
- Aprendizagem: ajuste de pesos iterativamente.

Fazer a configuração através de modelagem é um problema muito complicado. Através de aprendizagem temos o problema de já saber o que precisa ser codificado. Optamos, então, pela forma de configuração da rede através da sua equação de energia.

Utilizaremos a seguinte função:

$$\sum_{i=1}^{n} \Theta_i Si - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} S_i S_j$$

A variação de energia após a modificação de estado de um neurônio i é dada por:

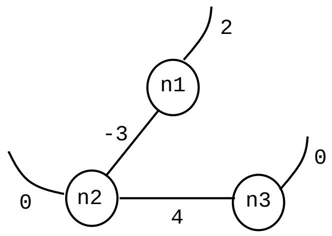

Figure 1: Net example

$$\Delta E = \Theta_i - \sum w_{ij} S_j$$

para todo vizinho j de i.

Por exemplo, se tivermos a equação de energia dada por:

$$E = 2S_1 + 3S_1S_2 - 4S_2S_3$$

teremos a representação em rede da figura .

Encontrar uma solução na rede significa resolver um problema de satisfabilidade, onde o máximo da equação de energia deve ser 1 (fórmula é satisfatível). Geralmente o procedimento de prova é realizado através de refutação, donde utilizaremos uma rede cuja equação de energia cresce a medida que a não satisfabilidade cresce.

Podemos definir:

- H(V) = 1, onde V é a constante verdadeira.
- H(F) = 0, onde F á constante falsa.
- H(p) = p, onde p é uma constante proposicional.
- H(p∧q) = p.q.
- H(¬p) = $1 - p$.
- H(p∨q) = H(p) + H(q) - H(p∧q).

Como encontrar a equação de energia a partir das definições lógicas acima? Convertemos a fórmula a ser provada para forma clausal: conjunção de disjunções. Como estamos utilizando um método de prova por refutação negamos o literal a ser resolvido. Temos então que o mínimo de E corresponde ao máximo da satisfabilidade.

$$E^* = H(\neg\phi) = H(\vee\neg(\wedge\phi_i)) =$$

$$= H(\vee_i\neg\phi_i) = H(\vee_i\neg(\vee_{p_{ij}})) =$$

$$= H(\vee_i \wedge_j \neg p_{ij})$$

consequentemente:

$$E = \sum_i H(\wedge_j\neg p_{ij})$$

Sistemas de inferência baseados em modelos conexionistas surgem como uma alternativa ao processamento paralelo de inferência que até então tem sido pesquisado para modelos de arquitetura convencionais [20, 21, 4, 5, 6, 17, 16, 11]. A rede de neurônios implementando um sistema de inferência representa um conjunto de regras que devem ser satisfeitas a partir de uma entrada do tipo consulta a um banco de dados, por exemplo. Novos valores (sinais) para os nós da rede são calculados através de uma equação matemática que rege o mecanismo de inferência. A rede entra em processo iterativo para a busca de soluções e tenta aprender através do conhecimento adquirido e regras de aprendizagem.

O processamento de Inferência por arquiteturas neuronais simétricas, no caso proposicional e de cálculo de predicados, pode ser feito através modelagem de Pinkas [12, 13], e por ARQ-PROPs [9], ARQ-PROP e ARQ-FOL [10]. Sistemas para Raciocínio Não-Monotônico e outras extensões podem ser encontrados em [2, 8, 14, 19].

References

[1] Ballard, D. H. Parallel Logical Inference and Energy Minimisation. Tech. Rep. Technical Report TR 142, Computer Science Dept. University of Rochester, Rochester, 1986.

[2] Derthick, M. Mundane reasoning by settling on a plausible model. In *Connectionist Symbol Processing*. Elsevier, 1990, pp. 107–158.

[3] Gregory, S. *Parallel Logic Programming in PARLOG*. Addison–Wesley, 1987.

[4] Gupta, G. *Multiprocessor Execution of Logic Programs*. Kluwer Academic Press, 1994. http://www.cs.nmsu.edu/lldap/.

[5] Gupta, G., Ali, K., Carlsson, M., and Hermenegildo, M. Parallel Logic Programming: A Survey. Research Report, ftp://pub.cs.nmsu.edu/pub/papers/lp/survey.ps.Z.

[6] Gupta, G., Hermenegildo, M. V., Pontelli, E., and Santos Costa, V. ACE: And/Or-parallel Copying-based Execution of Logic Programs. In *Proceedings of the Eleventh International Conference on Logic Programming* (Italy, June 1994).

[7] Hitz, M., and Kaltofen, E., Eds. *Proceedings of the Second International Symposium on Parallel Symbolic Computation, PASCO'97* (July 1997).

[8] Lima, P. M. V. Logical Abduction and Prediction of Unit Clauses in Symmetric Hopfield Networks. In *Artificial Neural Networks*, vol. 2. Elsevier, 1992.

[9] Lima, P. M. V. Resolution-Based Inference with Artificial Neural Networks. In *Workshop on Logic and Reasoning with Neural Networks at the International Conference on Logic Programming (ICLP-94), Santa Margherita* (1994). Workshop Proceedings can be retrieved at ftp.site: ftp.di.unipi.it, dir: pub/Papers/perso/ICLP-WS.

[10] Lima, P. M. V. Resolution-Based Inference with Artificial Neural Networks: Avoiding Cyclic Dependencies. In *Proceedings of the II Brazilian Conference on Neural Networks (CBRN-95), Curitiba, Brazil* (1995).

[11] Montelius, J., and Haridi, S. An evaluation of Penny: a system for fine grain implicit parallelism. In *Proceedings of 2nd International Symposium on Parallel Symbolic Computation [7]* (July 1997).

[12] Pinkas, G. Energy minimization and the satisfiability of propositional calculus. *Neural Computation 3 2* (1991), 282–291.

[13] Pinkas, G. Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), Sydney* (1991), pp. 525–530.

[14] Pinkas, G. Propositional non-monotonic reasoning and inconsistency in symmetric neural networks. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (1991), pp. 525–530.

[15] Pinkas, G. *Logical Inference in Symmetric Neural Networks*. PhD thesis, Sever Institute of Technology, Washington University, Saint Louis, In preparation, 1992.

[16] Santos Costa, V., Bianchini, and Dutra, I. C. Parallel Logic Programming Systems on Scalable Multiprocessors. In *Proceedings of the 2nd International Symposium on Parallel Symbolic Computation, PASCO'97 [7]* (July 1997), pp. 58–67.

[17] Santos Costa, V., Bianchini, R., and Dutra, I. C. Evaluating the impact of coherence protocols on parallel logic programming systems. In *Proceedings of the 5th EUROMICRO Workshop on Parallel and Distributed Processing* (1997), pp. 376–381. Also available as technical report ES-389/96, COPPE/Systems Engineering, May, 1996.

[18] Shapiro, E., Ed. *Concurrent Prolog–Collected Papers*. MIT Press, 1987.

[19] Vingralek, R. A Connectionist Approach to Finding Stable Models and Other Structures in Non-Monotonic Reasoning. In *Logic Programming and Non-Monotonic Reasoning*. MIT Press, 1993.

[20] Warren, D. H. D. An Abstract Prolog Instruction Set. Technical Note 309, SRI International, 1983.

[21] Yang, R., Santos Costa, V., and Warren, D. H. D. The Andorra-I Engine: A parallel implementation of the Basic Andorra model. In *Proceedings of the Eighth International Conference on Logic Programming* (1991), MIT Press, pp. 825–839.