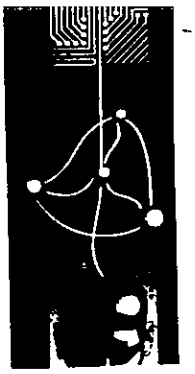

1° Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajubá
Itajubá, 24 a 27 de outubro de 1994

Requisitos de um Ambiente para Simulação de Redes Neurais Artificiais [1]

Patrícia Duarte de Lima Machado
Edson Costa de Barros Carvalho Filho

UFPE - Universidade Federal de Pernambuco
Departamento de Informática
Cx 7851, 50.732-970, Recife, PE, Brasil
Telefone: (081)271-8430, Fax: (081) 271-4925
Internet: {pdlm, ecdbcf}@di.ufpe.br

Sumário

Este artigo apresenta alguns requisitos a serem considerados na construção de ambientes para simulação de redes neurais artificiais. O objetivo é identificar, com um alto nível de abstração, os objetivos, as características e as restrições, tal que possam ser analisadas por futuros usuários do ambiente e que sirvam como base para o processo de desenvolvimento do software. Para elaboração dos requisitos, foram realizadas entrevistas com pesquisadores de redes neurais, iniciantes e especialistas, e foi estudado um grupo representativo dos simuladores atuais. Os requisitos definidos estão sendo usados no desenvolvimento do ambiente EASY.

Abstract

This paper shows some requirements to be considered in the construction of artificial neural networks simulation environments. The goal is identify, at a high abstraction level, objectives, features and restrictions, which may be analysed by users and may be used at the software development process. To elaborate the requirements, novice and expert neural network researchers were consulted and a representative group of actual simulators were studied. The requirements defined are been used at the environment EASY development.

[1] Este trabalho foi desenvolvido com o apoio do CNPq.

1 Introdução

Simulação em computador é uma opção comumente adotata por projetistas de redes neurais, visando especificar arquiteturas e verificar os seus comportamentos. A falta de uma notação formal que permita a realização de provas formais para verificação de propriedades é o maior fator que contribui para que a simulação continue sendo uma atividade dominante. Algumas notações já foram propostas [San94, Fie94], mas não possuem semantica formal. Mesmo que tal notação exista, sua aplicação, apesar de aparentemente mais confiável, torna-se difícil a medida que aumenta o número de componentes envolvidos no processo. Por esta razão, a simulação é vista como um importante aliado ao desenvolvimento das pesquisas em redes neurais [HBG90, HGR92].

Existem tres tipos básicos de implementação de redes neurais [Sim90]: *implementações em software*, executadas em um computador convencional, *implementações em hardware*, qualquer implementação eletronica com único propósito de processar um ou mais modelos de redes neurais, e *implementações óticas*, construídas utilizando componentes óticos.

Implementações em hardware dedicado, utilizando técnicas VLSI, visam dar mais velocidade a execução das redes. Uma desvantagem é que a capacidade de modificar o modelo fica comprometida. Uma outra desvantagem é que o tempo para desenvolver um sistema deste tipo é tipicamente da ordem de meses. Contudo, seu uso é adequado às situações nas quais o modelo a ser usado estiver completamente definido e testado [HGR92].

Implementações em software perdem em velocidade com relação às implementações em hardware e óticas, mas ganham em flexibilidade para modificações e expansões do modelo escolhido. Normalmente, são feitas utilizando uma linguagem de programação de propósitos gerais como C, C++, Smalltalk, C++ Concorrente, entre outras. Apesar de flexível, apresenta como desvantagem o fato de que um software tem de ser desenvolvido sempre que um paradigma tiver que ser simulado para uma dada aplicação. Se apenas a etapa de programação

for considerada no processo de desenvolvimento do software, então o produto resultante não deverá ter a qualidade e confiabilidade necessária, visto que nenhuma metodologia, técnica ou método é adotado para evitar a presença de definições ambíguas, inconsistentes e incompletas. Deverá também ser de difícil manutenção. A possibilidade de ser reutilizado por outros projetistas é praticamente nula, devido a dificuldade de se entender um código em uma linguagem de programação convencional. Outro problema com este tipo de simulação é que o projetista perde um valioso tempo com a implementação do software. Este tempo poderia ser utilizado em outros estudos.

Um simulador de rede neural pode ser definido como um pacote de software criado com o propósito específico de reduzir o tempo e o esforço envolvidos na solução de um problema usando redes neurais [Mur94]. Sistemas deste tipo devem permitir que pesquisadores possam testar novas idéias com um esforço mínimo e prover o processamento necessário a investigação de modelos complexos. Os simuladores de redes neurais podem ser classificados em tres tipos: *programa demonstrativo*, cujo objetivo é ilustrar um paradigma de rede especfico e seu comportamento; *simulador de propósito específico*, que possibilita a definição de um número variado de aplicações usando um conjunto particular de paradigmas de rede; e *ambiente de simulação*, que dá suporte a uma larga classe de paradigmas de rede bem como a definição e investigação de novos.

A utilização de um simulador oferece muitas vantagens sobre a construção de um *software* específico. Geralmente, permite que o usuário especifique a rede com um alto nível de abstração e de uma forma mais compacta, ao contrário do que ocorreria se o modelo tivesse que ser construído usando uma linguagem de programação. Muitos pesquisadores de redes neurais não são especialistas em computadores. Portanto, podem não dispor de tempo ou da habilidade necessária a construção de software.

O grande problema dos simuladores atuais é que estes não dão suporte a realização de um experimento completo, ou seja, não cobrem todas as etapas do processo de simulação de uma rede neural. Além disso, na grande maioria dos casos, possuem uma interface de difícil utilização e, portanto, consomem quase o mesmo tempo que o pesquisador levaria para construir seus próprios programas. Por, este motivo, na prática, estes simuladores são mais utilizados por iniciantes na área, os quais realizam pequenos experimentos a fim de aprender e explorar os diversos paradigmas.

EASY (*An [E]nvironment for [A]rtificial Neural [SY]stems Simulation*) [MFMG94] é um ambiente de simulação cujo principal objetivo é dar suporte a realização de um experimento completo usando redes neurais artificiais. Especificações Formais e Orientação a Objetos foram adotadas em seu processo de desenvolvimento visando assegurar qualidade, confiabilidade e reusabilidade ao produto final. A especificação formal utilizada pode ser encontrada em [GM93, MMFG94].

As seções seguintes apresentam um modelo para o processo de simulação de redes neurais e a definição de requisitos. Por fim, são dadas as conclusões sobre o trabalho realizado.

2 Um Modelo para o Processo de Simulação de Redes Neurais

Assim como existem modelos para o processo de desenvolvimento de software os quais especificam as atividades a serem seguidas para a geração de um produto, existe também a necessidade da definição de um modelo para o processo de simulação de uma rede neural que seja largamente aceito e que possa ser comumente adotado. Esta necessidade deve-se ao fato de que a computação neural é também uma abordagem para o processamento de informações destinada a aplicação em problemas para os quais algoritmos e regras não são conhecidos ou são, porém os softwares para implementá-los são caros, consomem muito tempo ou são inconvenientes para desenvolver [HN90].

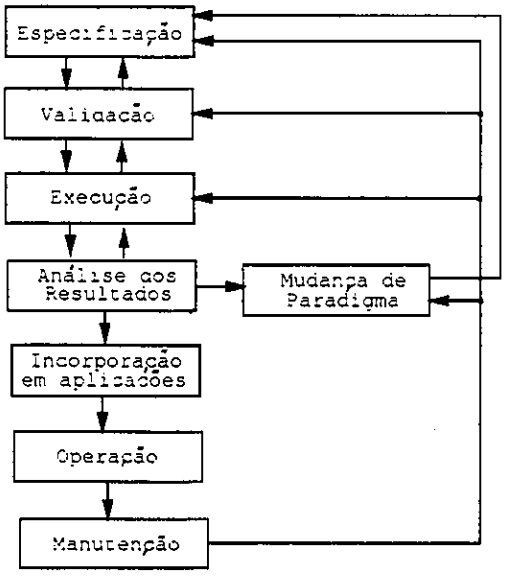

Figura 1: Um modelo para o ciclo de vida de aplicações usando redes neurais.

A concepção de um ambiente para simulação de redes neurais deve levar em consideração a existencia de um modelo para o processo, a fim de

poder fornecer o suporte necessário a realização do mesmo. Somente a adoção de um modelo pode conduzir a completude do ambiente, ou seja, o fornecimento de todo o suporte necessário. Caso nenhum modelo seja adotado, os usuários poderão ter que utilizar outras ferramentas a fim de complementar as etapas não cobertas pelo software de simulação escolhido. Na prática, muitos simuladores neurais se concentram em prover eficiência em apenas algumas das etapas do processo [Mur94].

[Mur94] sugere um conjunto de etapas que, na maioria dos casos, devem ser seguidas para a simulação de uma rede neural. Tais etapas são as seguintes:

1. **Especificação**. Consiste na tradução de um problema em uma representação de rede neural. Isto implica em escolher um conjunto de padrões a ser apresentado a rede, o subconjunto que melhor representa o domínio da aplicação, o paradigma a ser adotado, e a definição da arquitetura da rede, isto é, a quantidade, distribuição e conectividade dos neurônios.

2. **Validação**. Nesta etapa, é feita uma validação das escolhas efetuadas na etapa 1, tais como: a arquitetura da rede, o subconjunto escolhido como representativo do domínio da aplicação, entre outros.

3. **Execução da Simulação**. Dependendo do paradigma escolhido, consiste na aplicação dos algoritmos apropriados para efetuar a aprendizagem da rede.

4. **Análise dos Resultados**. Nesta etapa, são analisadas as saídas geradas pela rede após o processo de aprendizagem, o nível de generalização e a capacidade de abstração e outros parâmetros de interesse de acordo com o paradigma escolhido.

5. **Escolha de um outro Paradigma**. O paradigma que está sendo utilizado pode ser extendido para incluir características adicionais que melhor se adequem a solução do problema em questão ou mesmo um novo paradigma pode ser escolhido. Esta etapa ocorre com muita freqüência, principalmente se um novo problema estiver sendo tratado. A execução desta etapa implica num retorno obrigatório à 1.

6. **Incorporação em Aplicações Reais**. Consiste na geração de programas de aplicação ou esquemas para implementação em hardware.

Estas etapas podem ser interpretadas como a tentativa de definição de um modelo para o processo de simulação de redes neurais. A figura 1 apresenta um modelo mais completo para o ciclo de vida de aplicações usando redes neurais, onde é incluída também as fases de uso ou operação e manutenção da rede. Assim como no modelo *waterfall* [Som89], que é um modelo de desenvolvimento de software, o retorno a uma etapa anterior é previsto. A manutenção é feita para efetivar mundanças no domínio da aplicação ou corrigir erros.

3 Definição de Requisitos

A definição de requisitos é uma descrição em linguagem natural dos requisitos de um software. Consiste em uma descrição dos conceitos do sistema e a justificativa para cada decisão tomada. Existem dois tipos básicos de requisitos: requisitos funcionais e não-funcionais. Os requisitos funcionais definem a forma como o software deve operar e os requisitos não-funcionais definem os seus objetivos, características e restrições.

Após a realização de um estudo sobre alguns simuladores e de entrevistas com pesquisadores da área, foram definidos alguns dos requisitos básicos que um ambiente de simulação de redes neurais deve possuir a fim de que possa ser amplamente aceito e utilizado. Vale ressaltar que a completude destes requisitos fica comprometida devido ao grande número e a grande diversidade de usuários. Segundo [Som89] é muito difícil formular uma especificação definitiva para sistemas de software de grande porte. Assim, a medida que novos usuários forem sendo consultados estes requisitos poderão mudar e novos requisitos poderão ser identificados.

Neste trabalho, são apresentados apenas os requisitos não-funcionais por falta de espaço. Os requisitos funcionais podem ser encontrados em [Mac94]. Os requisitos não-funcionais identificados são os seguintes:

1. **Múltiplos Paradigmas**. Devem ser fornecidos pelo ambiente diferentes paradigmas de redes neurais, incluindo diferentes tipos de neurônios, arquiteturas e algoritmos de aprendizagem, para que o usuário possa escolher o modelo mais adequado a solução do seu problema. Este é um dos objetivos de projeto do SESAME [LSTW92].

2. **Diferentes Aplicações**. Diferentes tipos de padrões devem poder ser utilizados pelo sistema a fim de possibilitar a definição de diversos tipos de aplicações. Este é um outro objetivo do SESAME.

3. **Flexibilidade para Incorporação de Novos Paradigmas**. Devido a constante evolução da teoria de redes neurais, é imprescindível que um ambiente de simulação ofereça

facilidades para a incorporação de novas arquiteturas, modelos de neuronio e algoritmos de aprendizagem. Esta característica está intimamente relacionada com a evolutibilidade do software. A reutilização de definições é de grande importância para que pesquisadores tornem-se aptos a testar novas idéias com rapidez e com um mínimo de esforço. A orientação a objetos tem sido utilizada como maneira preferencial para a modelagem de redes neurais, pois promove a modularidade e a reusabilidade [HBG90, LSTW92, FAT93].

4. **Execução em Hardware Paralelo**. Redes neurais possuem um carater inerentemente paralelo e é crescente o número de implementações que tiram proveito da execução paralela para diminuir o tempo de simulação das redes. O simulador parsimANNs [MWGH92] possibilita execução em hardware paralelo.

5. *Interfaces* **de fácil entendimento e utilização**. Devem permitir que o usuário possa utilizar o ambiente sem ter que se preocupar com os seus detalhes operacionais, ou ter que perder muito tempo lendo manuais exaustivos com comandos para interação.

6. *Interfaces* **Consistentes**. Uma interface consistente é aquela em que quaisquer dois comandos, opções de menu ou botões que possuam uma identificação com a mesma sintaxe devem produzir um comportamento análogo. Esta é uma característica encontrada no ambiente de programação SmallTalk [Sys90]. Uma vez tendo aprendido os comandos de uma dada ferramenta do ambiente, em qualquer outra ferramenta onde forem encontrados comandos com a mesma sintaxe, estes serão facilmente compreendidos pelo usuário.

7. *Interfaces* **Reusáveis**. A reusabilidade é um fator primordial para viabilizar a expansão do sistema. Por interfaces reusáveis podemos entender, por exemplo, que uma interface para dar suporte a uma atividade de um dado paradigma deve ser projetada, tal que, um novo paradigma incorporado no ambiente reutilizando as definições do primeiro possa também reutilizá-la na definição de sua própria. Reutilização de interfaces é discutido em [ES92].

8. **Interfaces com diferentes visões de usuário**. De acordo com o tipo de usuário, um maior ou menor número de opções deve ser apresentado. Por exemplo, para um iniciante, apenas algumas opções devem ser fornecidas, devido ao seu pouco conhecimento. Porém para um usuário especialista tanto na área de redes neurais quanto na própria utilização do ambiente, um maior número de opções deve ser apresentado.

9. **Especificação de arquiteturas selecionando objetos da interface**. O objetivo é facilitar o trabalho do projetista de uma rede neural, evitando que este tenha que construir programas ou aprender uma seqüência de comandos para especificar as arquiteturas com as quais deseja trabalhar, reduzindo, significativamente, o tempo dispendido para esta tarefa. Na interface ANNview [MWGH92], uma parte das definições são efetuadas usando *mouse* com operações, *point, clic e drag* em *icons* gráficos. Assim, o usuário não precisa ser especialista em programação para definir novos paradigmas ou testar os pré-existentes. Os simuladores RCS [GLM88] e UCLA-SFINX [MS92] adotam a linguagem de programação *C* para a especificação de arquiteturas.

10. **Especificação de novas funções de ativação/aprendizagem utilizando uma linguagem declarativa com sintaxe e semântica formais**. Linguagens procedurais induzem a especificação de "como" deve ser implementado, aumentando o trabalho do projetista. Linguagens declarativas possuem um alto nível de abstração, possibilitando a descrição do "que" deve ser implementado. Linguagens formais são baseadas em matemática e, por este motivo, evitam definições ambígüas e inconsistentes. A maioria dos simuladores utilizam linguagens procedurais. No simulador NeuroGraph [Wil93], funções são especificadas usando expressões matemáticas.

11. **Execução** *foreground* × *background*. A execução da rede em *foreground* é importante, pois possibilita que o usuário possa acompanhar o processo. Porém, uma vez aprovados todos os parâmetros a serem utilizados, a execução em *background* se torna mais adequada, visto que, para redes grandes, o tempo de execução é, normalmente grande, e, portanto a execução em *background* libera o usuário para executar outras atividades. Os simuladores SNNS [ZMH+93] e SESAME [LSTW92] são exemplos de simuladores que fornecem as duas opções.

12. **Interação durante a execução em** *foreground*. O usuário quando executando uma simulação em *foreground*, deve ser capaz de acompanhar e controlar a evolução do processo, tal que possa interrompê-lo para alterar alguns parametros. Este é o caso do SESAME [LSTW92].

13. **Salvar as informações de status**. Todas as definições dos usuários devem poder ser guar-

das em um arquivo para que em uma posterior sessão de trabalho, estas possam ser utilizadas sem que o usuário precise especificá-las novamente. No Aspirin [Lei92], o treinamento pode ser interrompido e o estado da rede pode ser salvo para ser carregado posteriormente, dando continuidade ao processo.

14. **Auto-inicialização**. Todas as definições necessárias a execução de uma atividade devem ser automaticamente inicializadas pelo sistema para evitar que resultados inconsistentes sejam gerados. Isto é necessário caso o usuário forneça uma especificação incompleta. O SESAME [LSTW92] trata este problema.

15. *Help on-line* **como um hipertexto**. Um *Help* com estrutura não-linear, ou seja, na forma de um hipertexto [SFAM92], deve estar disponível para qualquer ferramenta do ambiente. Ele deve ser sensível ao contexto da ferramenta a partir da qual for invocado, mas, por outro lado, deve permitir que por navegação, o usuário tenha acesso a qualquer informação sobre qualquer ferramenta do sistema. Entre as informações que devem estar disponíveis, temos a semântica dos botões e *menus* das ferramentas, um guia com uma possível sequencia de passos a serem seguidos para concretização de uma tarefa, descrição teórica das funções executadas pela ferramenta (por exemplo, a especificação matemática de um algoritmo de aprendizagem) e a teoria completa sobre o paradigma escolhido. Estas duas últimas informações seriam de grande importância para iniciantes na área de redes neurais.

16. **Anotações** Todas as ferramentas do ambiente devem permitir que anotações particulares sejam feitas pelo projetista, para que este possa registrar, a cada etapa, comentários próprios sobre as decisões que forem sendo tomadas. Estas anotações devem ser guardadas juntamente com as informações de *status* da ferramenta e as opções feitas pelo projetista. Um dos problemas do SESAME [LSTW92] é que este não armazena os comentários do projetista juntamente com as especificações de experimentos.

17. **Customização do Ambiente**. As opções de visualização gráfica e textual fornecidas em todas as ferramentas devem ser opcionais, para que o usuário possa escolher a que mais lhe convier e para que não haja um sobrecarga cognitiva com um grande número de informações sendo apresentadas ao mesmo tempo. O UCLASFINX [MS92] permite que o usuário defina seu ambiente de simulação.

4 Conclusões

O principal objetivo deste trabalho é identificar, com um alto nível de abstração, os requisitos de um ambiente para simulação de redes neurais, tal que seja amplamente aceito e utilizado. A importância da aquisição destes requisitos é justificada, quando se leva em consideração o grande número e diversidade de usuários deste tipo de sistema. Muitos estudos têm mostrado que erros encontrados em fases mais avançadas do processo de desenvolvimento de software são mais caros para correção do que os detectados no início do processo. A definição dos requisitos permite que os objetivos sejam estabelecidos e investiga as necessidades dos usuários e o domínio da aplicação.

Os requisitos apresentados neste trabalho podem ser aplicados no processo de desenvolvimento de qualquer ambiente para simulação de redes neurais artificiais. Em particular, está sendo utilizado no desenvolvimento do ambiente EASY como base para a especificação formal de suas propriedades.

Referências

[ES92] Raimund K. Ege and Christian Stary. Designing maintainable, reusable interfaces. *IEEE Software*, pages 24–32, november 1992.

[FAT93] L. Fuentes, J. F. Aldana, and J. M. Troya. Urano: An object-oriented artificial neural network simulation tool. Em J. Mira, J. Cabestany, and A. Prieto, editors, *New Trends in Neural Computation: International Workshop on Artificial Neural Networks - IWANN*, pages 364–369, Sitges, Spain, june 1993.

[Fie94] E. Fiesler. Neural network classification and formalization. To be published in "Computer Standards & Interfaces, volume 16, special issue on Neural Network Standards, John Fulcher, editor. North-Holland, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1994, ISSN 0920-5489, 1994.

[GLM88] Nigel H. Goddard, Kenton J. Lynne, and Toby Mintz. Rochester connectionist simulator. Relatório Técnico 233, University of Rochester, Computer Science Department, Rochester, New York, 14627, march 1988.

[GM93] H. M. Gomes and P. D. L. Machado. Especificando Redes Neurais Artificiais

em MooZ. Relatório técnico, Universidade Federal de Pernambuco, Departamento de Informática. Recife - PE. 1993.

[HBG90] G. L. Heileman, H. K. Brown, and M. Georgiopoulos. Simulation of artificial neural network models using an object-oriented software paradigm. Em *Proceedings of the International Joint Conference on Neural Networks*, pages II–133–II–136, Washington, DC, 1990.

[HGR92] G. L. Heileman, M. Georgiopoulos, and W. D. Roome. A general framework for concurrent simulation of neural network models. *IEEE Transactions on Software Engineering*, 18(7):551–562, july 1992.

[HN90] Robert Hecht-Nielsen. *Neurocomputing*. Addison-Wesley Publishing Company, Inc. 1990.

[Lei92] R. R. Leighton. *The Aspirin/ MIGRAINES Neural Network Software - User's Manual*. MITRE Corporation, 1992.

[LSTW92] A. Linden, Th Sudbrak, Ch Tietz, and F. Weber. An object-oriented framework for the simulation of neural nets. Em C. L. Giles, S. J. Hanson, and J. D. Cowan, editors, *Advances in Neural Information Processing Systems 5*. CA: Morgan Kaufmann Publishers, San Mateo, 1992.

[Mac94] Patrícia D. L. Machado. Proposta de um ambiente para modelagem, simulação e análise de redes neurais artificiais. Tese de Mestrado, Universidade Federal de Pernambuco, Departamento de Informática, Recife - PE, Brasil. 1994.

[MFMG94] P. D. L. Machado, E. C. D. B. Carvalho Filho, S. R. L. Meira, and H. M. Gomes. EASY - an [E]nvironment for [A]rtificial neural [SY]stems simulation. To be published at Fourth Irish Neural Network Conference - INNC'94, Dublin, Ireland, 1994.

[MMFG94] P. D. L. Machado, S. R. L. Meira, E. C. D. B. Carvalho Filho, and H. M. Gomes. Especificando redes neurais artificiais em MooZ. To be published at XX Conferencia Latino Americana de Informatica - PANEL'94, Mexico, 1994.

[MS92] Edmond Mesrobian and Josef Skrzypek. A software environment for studying computacional neural systems. *IEEE Transactions on Software Engineering*, 18(7):575–589, july 1992.

[Mur94] Jacob M. J. Murre. Neurosimulators. To be published in the "HandBook of Brain Research and Neural Networks", M. A. Arbib, MIT Press, 1994.

[MWGH92] Harley R. Myler, Arthur R. Weeks, Randall K. Gillis, and Gary W. Hall. Object-oriented neural simulation tools for a hypercube parallel machine. *Neurocomputing*, 4(5):235–248, 1992.

[San94] D. A. Santos. MOdelo Formal de Especificação Universal para Redes Neurais - MOFEU. Tese de Mestrado, Universidade Federal de Pernambuco, Departamento de Informática, Recife - PE, Brasil, 1994.

[SFAM92] A. C. Salgado, D. Fonseca, E. S. Albuquerque, and S. R. L. Meira. *Sistemas Hipermídia: Hipertexto e Banco de Dados*. VIII Escola de Computação, Gramado, RS, 1992.

[Sim90] Patrick K. Simpson. *Artificial Neural Systems: Foundations, Paradigms, Aplications and Implementations*. Pergamon Press, Inc. 1990.

[Som89] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, 3 edition, 1989.

[Sys90] ParcPlace Systems. *Objectworks.\ Smalltalk - User's Guide*. 1550 Plymouth Street, Mountain View, California 94043, 1990.

[Wil93] Peter Wilke. Simulation of neural networks in a distributed computing environment using neurograph. Em J. Mira, J. Cabestany, and A. Prieto, editors, *New Trends in Neural Computation: International Workshop on Artificial Neural Networks - IWANN*, pages 394–398, Sitges, Spain, june 1993.

[ZMH+93] A. Zell, N. Mache, R. Hübner, G. Mamier, M. Vogt, K. Herrmann, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, S. Döring, and D. Posselt. SNNS: Sttugart Neural Network Simulator - User Manual. Relatório Técnico 3, University of Stuttgart, Institute for Parallel and Distributed High Performance Systems, Breitwiesenstr. 20-22, 70565 Stuttgart, Fed. Rep. of Germany, 1993.