



Exacta

ISSN: 1678-5428

exacta@uninove.br

Universidade Nove de Julho

Brasil

Toscano, Wagner; Lobo Lustosa Cabral, Eduardo
Contextualização de redes neurais recorrentes
Exacta, vol. 4, núm. Esp, novembro-special, 2006, pp. 65-66
Universidade Nove de Julho
São Paulo, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=81009907>

- ▶ Como citar este artigo
- ▶ Número completo
- ▶ Mais artigos
- ▶ Home da revista no Redalyc

re&alyc.org

Sistema de Informação Científica

Rede de Revistas Científicas da América Latina, Caribe , Espanha e Portugal
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

Contextualização de redes neurais recorrentes

Wagner Toscano^{1,2}, Eduardo Lobo Lustosa Cabral²

¹Uninove, Departamento de Ciências Exatas, São Paulo, SP [Brasil].

²USP, Escola Politécnica, Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, São Paulo, SP [Brasil].
wtoscano@usp.br

Atualmente, os estudos sobre redes neurais artificiais (RNAs) convergem para o conceito de RNAs recorrentes (RNARs), pela facilidade de implementação e redução de componentes denominados neurônios e pela agilidade com que as redes recorrentes podem ser treinadas em relação às neurais em geral. Soma-se a esses motivos a similaridade existente com as redes biológicas.

Sobre o paradigma da inteligência artificial conexista, acredita-se que, construindo-se um sistema que simule a estrutura do cérebro, esse sistema apresentará inteligência, ou seja, será capaz de aprender, assimilar, errar e aprender com seus erros (HOCH, 1997).

Em geral, a inteligência consiste em um processo de raciocínio composto de mecanismos de captura de dados e operações de validação desses dados. Nessas operações os instrumentos incluem modos de inferência.

A inferência comprehende uma operação intelectual por meio da qual se afirma a verdade de uma proposição, em decorrência de sua ligação com outras proposições já reconhecidas como verdadeiras, formando um encadeamento lógico e argumentativo. Dos três modos de inferência – abdutiva, indutiva e dedutiva –, o único que pode produzir exatidão na conclusão é o da inferência dedutiva. No entanto, essa exatidão falha se não for obtida a descrição dos estados do ambiente analisado, isto é, tendo como premissa que os estados de um ambiente são contínuos, sofrendo alterações no tempo e no espaço, a representação

do ambiente torna-se complexa. Para contornar essa complexidade, a inferência dedutiva recorre a um *snapshot* do ambiente, por meio de restrições. Dessa forma, aceita-se como resultado de raciocínio uma solução próxima ao ideal, porém não 100% exata. Essa “inexatidão” nos resultados está alinhada ao conceito da inteligência artificial conexista, o qual propõe o raciocínio por aproximação, utilizando a rede neural artificial (RNA) como uma das ferramentas.

A RNA tem como base a estrutura dos neurônios biológicos e é composta de dispositivos chamados de neurônios artificiais que, apesar de não serem um substituto para os biológicos, atendem, como solução, a diversos problemas, principalmente aos que necessitam de soluções em tempo real (KRÖSE; SMAGT, 1996).

O desempenho de uma RNA depende dos parâmetros que a compõem, tais como bias, *threshold*, pesos; da estrutura projetada, se possui *layer* “escondida” ou não, e quantas; do algoritmo de treinamento e, principalmente, de sua aplicação.

Os algoritmos de treinamento são fundamentados nos conceitos-chave do aprendizado de Hebbian e na retropropagação (*backpropagation*) do erro. A técnica de aprendizado denominada *backpropagation* é relativamente simples de ser implementada quando existem poucos neurônios; no entanto, há dificuldade de evitar “mínimos locais” (oscilação na solução).

Com relação às *layers*, quanto maior a quantidade, maior é a capacidade de aprendizado e



maior a precisão com que serão delimitadas as regiões de decisão. No entanto, maior quantidade de *layers*, além de comprometer o tempo de treinamento, quando utilizada a *backpropagation*, causa o efeito *overfitting*, isto é, excesso de neurônios que impossibilita generalizar um exemplo. Além disso, RNAs com grande número de *layers* exigem complexidade de implementação.

Mesmo com esses problemas, uma RNA chega a um resultado mais rápido do que os algoritmos específicos otimizados.

Um tipo de RNA que minimiza o tempo de treinamento e a complexidade de implementação é chamado de RNA Recorrente (RNAR), ou “rede associativa”. Essa denominação é dada à categoria das RNAs dinâmicas, com representação do estado temporal, isto é, nelas podem ser acoplados elementos de atraso-unitário que, em conjunto com as características não-lineares dos próprios neurônios artificiais, fazem a rede operar em um domínio não-linear (UNADKAT; CIOCOIU; MEDSKER, 2001).

A estrutura de uma RNAR é composta de conexões, com pelo menos um ciclo de realimentação entre os neurônios, e assemelha-se à estrutura da rede neural *feedforward*, mas diferencia-se no comportamento e na forma de treinamento.

As RNARs aprendem rapidamente, necessitando de poucas iterações de treinamento (menos de mil), além de precisarem de menos parâmetros e fornecerem melhor generalização.

Pelo simples fato de uma RNAR não possuir *layers* explícitas (pois um ou todos os neurônios podem ser entradas e saídas), a implementação se torna simples e reduzida. Como exemplo, supondo uma RNA para solução do problema XOR,

seriam necessários três neurônios. Desses neurônios, dois formariam a *layer* de entrada e um a *layer* de saída. No entanto, em uma RNAR, para a solução do XOR, seriam necessários apenas dois neurônios.

Dependendo da estrutura da RNAR, o treinamento pode ser simples (XOR) ou complexo. Nos casos complexos, o problema no treinamento é semelhante aos problemas não recorrentes, encontrados em RNA.

Visando à motivação na redução de neurônios e à implementação características não-lineares, está sendo desenvolvido um sistema genérico de criação e implementação de RNAR, com treinamento fundamentado em algoritmo genético.

Resultados empíricos foram obtidos, empregando-se os conceitos em uma simulação no aplicativo Matlab, com efeitos satisfatórios, principalmente no que se refere à implementação.

O próximo passo consiste em desenvolver um sistema em ambiente Java, que resulte, além de um rápido treinamento, numa RNAR com reduzida quantidade de neurônios.

Referências

HOCH, H.; SCHMIDHUBER, S. J. Long short-term memory. *Neural Computation*, La Jolla, v. 9, n. 8, p. 1.735-1.780, 1997.

KRÖSE, B.; SMAGT, P. van der. *An introduction to neural networks*. 8. ed. Amsterdam: University of Amsterdam, 1996.

UNADKAT, S. B.; CIOCOIU, M. M.; MEDSKER, L. R. Introduction. In: MEDSKER, L. R.; JAIN, L. C. (Ed.). *Recurrent neural networks: design and applications*. 1. ed. Londres: CRC, 2001. p. 1.

Para referenciar este texto

TOSCANO, W.; CABRAL, E. L. L. Contextualização de redes neurais recorrentes. *Exacta*, São Paulo, v. 4, n. especial, p. 65-66, 25 nov. 2006.