

Neurocomputação na Estimativa do Parâmetro de Hurst*

Danielo Gomes^{*}
dgomes@iup.univ-evry.fr

Nazim Agoulmine^{*}
nazim.agoulmine@iup.univ-evry.fr

Nelson Fonseca^{**}
nfonseca@dcc.unicamp.br

José Neuman de Souza^{***}
neuman@ufc.br

^{*}Université d'Evry

^{**}Universidade Estadual de Campinas - UNICAMP

^{***} Universidade Federal do Ceará - UFC

RESUMO

Este trabalho propõe um estimador de parâmetro de Hurst através do uso de Redes Neurais Artificiais (RNAs) para o tráfego auto-similar de redes de computadores. A ferramenta neural constitui uma solução alternativa aos algoritmos tradicionais e sua eficácia é avaliada mediante uma comparação com outros três estimadores estatísticos. Investiga-se também a influência de alguns elementos das RNAs – número de neurônios escondidos e função de ativação – no desempenho final. Os resultados numéricos comprovam a possibilidade de aplicação das redes neurais neste contexto de avaliação da dependência temporal de tráfego de dados.

Palavras-chave: Gerenciamento de redes de computadores, Redes neurais, tráfego auto-similar, parâmetro de Hurst.

ABSTRACT

In this work, an Hurst parameter estimator by using Artificial Neural Networks (ANNs) has been developed. The traffic for this analysis is generated by an algorithm and the results are compared with three others estimators, in order to evaluate the accuracy of the proposal. It is also investigated the influence of some elements of the neural nets' learning – hidden neurons and activation function– in the final performance. Simulation results show that the neural network approach can be used on the Hurst parameter estimation for data traffic.

Keywords: Network management, neural networks, self-similar traffic, Hurst parameter.

* Este trabalho foi realizado com recursos da UFC, RNP(PoP-CE), FUNCAP e CAPES/COFECUB.

1. Introdução

Com o crescente uso de aplicações multimídia em redes, torna-se necessário um adequado dimensionamento dos seus recursos, a fim de se prover as diferentes demandas deste tipo de aplicação. Assim sendo, os mecanismos de gerenciamento de tráfego multimídia têm sido alvo de grande atenção nos últimos anos. Para que estes mecanismos sejam efetivos na manutenção da Qualidade de Serviço (*Quality of Service*, ou QoS), deve-se entender a natureza do tráfego de redes. Constatou-se, recentemente, que diversos tipos de tráfego, tais como os tráfego de redes locais e de vídeo VBR, apresentam correlações de longa duração. Estes tráfegos podem ser modelados por processos auto-similares, tais como o processo movimento Browniano fracional (*fractional Brownian motion*, ou fBm). Dependências de longa duração têm um impacto significativo no dimensionamento de redes. Os multiplexadores sujeitos a processos com dependências de longa duração podem apresentar perdas maciças de pacotes. A diminuição da perda de pacotes não ocorre com o aumento substancial do tamanho do *buffer* nestes cenários.

Um processo auto-similares apresenta rajadas em diferentes escalas de tempo, ou seja, ao se observar um processo auto-similar em diferentes escalas de tempo, constata-se uma semelhança no padrão das amostras do processo. O parâmetro de Hurst, denotado por H , é uma métrica que caracteriza o grau de auto-similaridade de um processo. Este parâmetro varia no intervalo $[0,1]$ com os valores acima de 0,5 representando correlações positivas e os valores abaixo correlações negativas. Quanto maior o valor do parâmetro de Hurst, maior é o impacto das auto-correlações no desempenho do sistema. Além disso, pequenas variações no parâmetro de Hurst de um tráfego podem demandar mudanças significativas no dimensionamento dos elementos da rede, bem como podem requerer reações preventivas imediatas dos mecanismos de gerenciamento de tráfego.

Existem diversos estimadores estatísticos para o parâmetro de Hurst. Alguns deles, tais como o método R/S baseiam-se em interpretação visual. Outros estimadores constroem intervalos de confiança em torno do valor estimado. No entanto, para uma análise acurada, estes estimadores requerem uma grande amostragem. Por outro lado, em algumas situações, tais como a transmissão ao vivo de vídeo, não é possível estimar-se previamente o valor do parâmetro H do tráfego gerado. Assim sendo, necessita-se de estimadores acurados que possam ser usados em tempo real e que obtenham uma resposta precisa em um curto intervalo de tempo, a fim de que mudanças no valor do parâmetro de Hurst possam ser imediatamente reportadas ao controlador da rede.

O objetivo do presente trabalho é investigar a efetividade da estimação do parâmetro de Hurst através de redes neurais. A precisão da nossa proposta é verificada mediante uma comparação com outros três métodos estatísticos de estimativa de parâmetro H : a estatística R/S, o método de Higuchi e o estimador Abry-Veitch.

O restante deste artigo está organizado da seguinte forma: a Seção 2 introduz os aspectos das redes neurais e seus princípios, a Seção 3 caracteriza o tráfego auto-similar, a Seção 4 mostra o cenário do ambiente de experimentação. Os três estimadores estatísticos são brevemente descritos na Seção 5, o protótipo neural é explicado na Seção 6 e os resultados das simulações constam na Seção 7. Finalmente, o trabalho é concluído na Seção 8.

2. Redes Neurais Artificiais

Para várias tarefas simples e comuns do nosso dia-a-dia, é extremamente difícil projetar um algoritmo que as resolva. Semão vejamos: nossa percepção visual (imagem) e/ou auditiva (voz) das coisas no mundo ou mesmo o modo como o homem reconhece os

caracteres manuscritos sem antes nunca tê-los visto. A experiência humana está organizada em células neurais.

As técnicas usadas em bancos de dados convencionais são insuficientes para responder pela ampla diversidade de associações que o homem pode fazer. Tome, por exemplo, o caso do reconhecimento da escrita. Convencionalmente, a solução passaria pelo reconhecimento de cada letra: isto implica que sejam armazenadas todas as possíveis formas de uma pessoa escrever cada caractere. Claro que isso é impraticável, pois nada garante que no texto a ser reconhecido haja somente as formas armazenadas nesta base de dados.

A partir destas considerações, surge o ímpeto natural de se copiar o mais fielmente possível a arquitetura do cérebro para que se possa emular, também o mais fielmente possível, suas funções.

Uma Rede Neural Artificial (RNA) ou simplesmente RNA é um sistema formado por um grande número de elementos processadores simples (neurônios ou nós), altamente interconectados e baseados no modelo simplificado do neurônio. O procedimento é de adaptar a resposta fornecida pela RNA à experiência de situações passadas (treinamento), à semelhança do que acontece no cérebro humano. Neurocomputação é uma abordagem computacional das redes neurais (estruturas conexionistas ou paralelas distribuídas) para o processamento da informação [1].

As redes neurais são o paradigma e a neurocomputação é a sua implementação através de *software*. Este é o teor deste trabalho, como consta no próprio título, haja vista o uso de um simulador de RNAs [2]

Dentre as características mais relevantes das RNAs, a que torna essa abordagem mais atrativa em muitos aspectos é a capacidade dos modelos aprenderem informações do ambiente onde estão inseridas e a sua habilidade em modificarem seu desempenho em função deste aprendizado. O emprego de uma rede neural na solução de determinada tarefa passa necessariamente, antes de tudo, por uma fase de treinamento. O treinamento é o processo pelo qual a rede neural aprende. Efetua-se o aprendizado através do reajuste dos pesos de suas conexões. Para isto existem os algoritmos de aprendizagem: são regras iterativas. De fato, os pesos representam o conhecimento da rede ao término do processo de treinamento e o resultado final desse processo, chamamo-lo de aprendizado. Assim, o aprendizado é um processo pelo qual os parâmetros livres da rede neural são adaptados através de um contínuo estímulo do ambiente no qual ela está inserida [3]. Esta definição nos sugere a ocorrência de uma seqüência de três eventos durante o processo de aprendizagem (Figura 1).

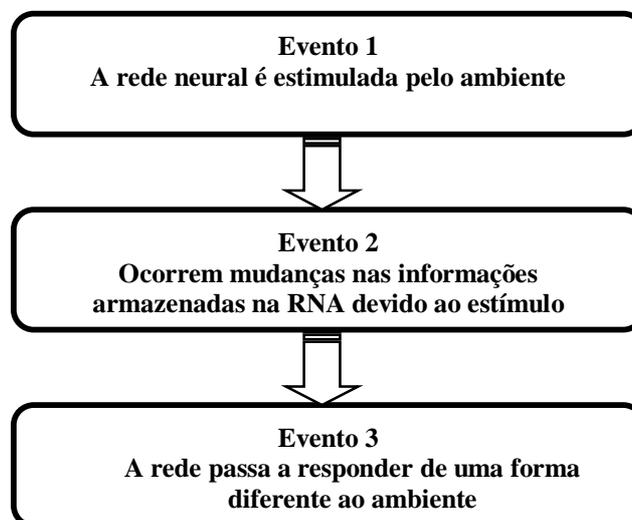


Figura 1: os eventos do aprendizado.

3. Tráfego Auto-similar

Para se visualizar melhor a idéia da auto-similaridade com o auxílio dos conjuntos de Cantor [4], a Figura 2 ilustra a construção destes conjuntos, os quais obedecem as seguintes regras:

- i) começam em um intervalo fechado $[0, 1]$, representado por um segmento de reta;
- ii) remove-se a terça parte central da linha;
- iii) para cada passo seguinte, elimina-se a terça parte central das linhas originadas do passo anterior.

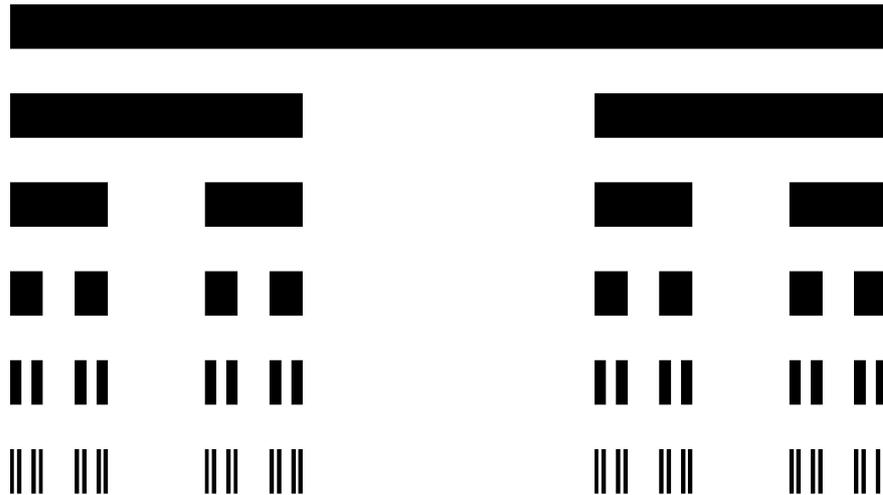


Figura 2: um conjunto de Cantor com cinco níveis de recursão.

Este processo recursivo pode ser definido matematicamente como segue. Considere S_i o conjunto de Cantor após i níveis de recursão. Logo,

$$S_0 = [0,1]$$

$$S_1 = [0,1/3] \cup [2/3,1]$$

$$S_2 = [0,1/9] \cup [2/9,1/3] \cup [2/3,7/9] \cup [8/9,1], \text{ e assim por diante.}$$

Ao considerar a reta de Cantor como uma linha de tempo, cada passo sucessivo divide a escala em três partes. Note que, a cada iteração, as porções esquerda e direita são réplicas exatas do conjunto inteiro do passo anterior. O conjunto de Cantor revela duas propriedades observadas nos fenômenos auto-similares [5]:

- i. o conjunto mantém uma certa estrutura em escalas arbitrariamente pequenas. Mesmo intensificando as subdivisões do conjunto, permanece ainda um padrão complexo de pontos separados por lacunas de vários tamanhos;
- ii. a estrutura se repete. Uma estrutura auto-similar contém pequenas réplicas de si mesma em todas as escalas.

Embora este exemplo seja simples, algumas idéias podem ser adquiridas a partir do seu estudo. Talvez a mais significativa característica, do ponto de vista de desempenho de redes de computadores, seja a persistência de grupos. Com suposições de tráfego baseadas em processos de Poisson, os agrupamentos ocorrem em curta duração (pequenas escalas de tempo), mas se descaracterizam em longas durações, i.e., as rajadas amortizam conforme os dados são agregados em escalas maiores de tempo [6].

Assim, pode-se projetar um sistema de servidores e filas com *buffers* na expectativa desta suavização do tráfego em longa duração. Isto implica que, pelo fato da “explosividade” (*burstiness*) diminuir durante um longo período, somente *buffers* de tamanhos modestos são necessários. Uma fila pode até ser formada em um curto espaço de tempo mas, a longo prazo, os *buffers* são esvaziados. Contudo, se o comportamento do tráfego apresenta rajadas dentro de rajadas (os grupos são agrupados), o tamanho das filas pode aumentar mais do que o previsto por um tráfego Poissoniano.

Estas considerações nos permitem afirmar que as análises tradicionais de filas, as quais supõem um fluxo Poissoniano de dados, podem incorrer em previsões não muito precisas para um fluxo de dados com características inerentes de auto-similaridade [7].

O parâmetro H , conhecido como parâmetro de Hurst é um indicador fundamental de auto-similaridade e estimar seu valor via redes neurais é o objetivo deste trabalho. Mais precisamente, H é uma medida da persistência e de longa duração (alcance) de um processo. Este trabalho adota a seguinte classificação [8] para a dependência temporal do processo mediante os valores do parâmetro H :

- $0,5 < H < 1$: processo de dependência de longo alcance (LRD) ou persistente;
- $H = 0,5$: processo de dependência de curto alcance (SRD);
- $0 < H < 0,5$: processo de dependência negativa ou anti-persistência.

Seja $x(t)$, com $t = 0, 1, 2, \dots$, um processo estocástico estacionário [9]. Para cada $m = 1, 2, \dots$, considere $x^{(m)}(k)$, $k = 1, 2, 3, \dots$, como sendo uma nova série temporal obtida a partir da seqüência estocástica original $x(t)$ sobre agrupamentos adjacentes e disjuntos de tamanho m .

Um processo X é chamado *exatamente auto-similar de segunda-ordem* com parâmetro $H = 1 - \beta/2$, $0 < \beta < 1$, se a sua função de auto-correlação é dada por [10]:

$$r^{(m)}(k) = r(k) = \frac{1}{2} \left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta} \right] \triangleq g(k), \quad m = 1, 2, \dots, \quad k = 1, 2, 3, \dots \quad (1)$$

e X é chamado *assintoticamente auto-similar de segunda-ordem* com parâmetro $H = 1 - \beta/2$, $0 < \beta < 1$, se

$$\lim_{m \rightarrow \infty} r^{(m)}(k) = \frac{1}{2} \left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta} \right] \triangleq y(k), \quad k = 1, 2, 3, \dots \quad (2)$$

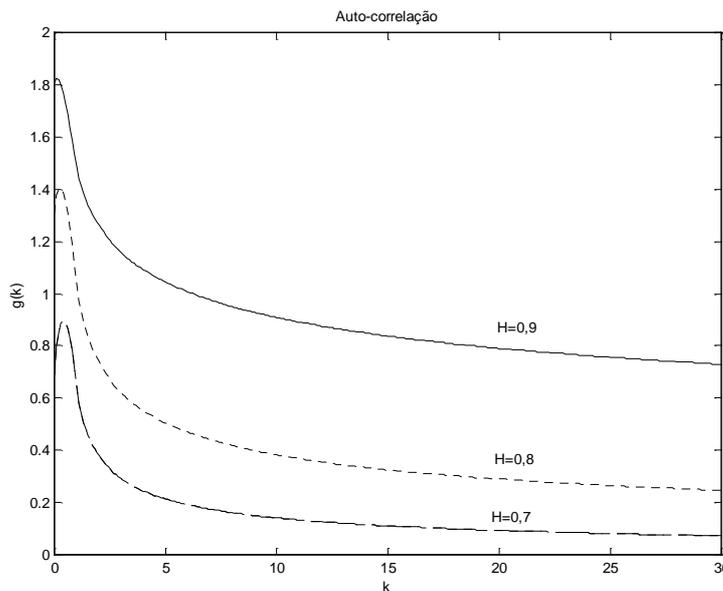


Figura 3: função de auto-correlação de um processo exatamente auto-similar de segunda-ordem com parâmetro de Hurst $H = 1 - \beta/2$.

A Figura 3 mostra a função de auto-correlação $g(k)$ (Equação (1)) como função de k para três valores de β (0,2;0,4 e 0,6). Estes valores de β representam conforme a equação $H = 1 - \beta/2$, respectivamente $H = 0,9$; 0,8 e 0,7.

Matematicamente, a auto-similaridade manifesta-se em um número de modos equivalentes:

- (i). a variância da média amostral decresce mais lentamente do que a variância do tamanho da amostra (*variância decaindo lentamente*), i.e., $var(X^{(m)}) \sim am^{-\beta}$, com $m \rightarrow \infty$, com $0 < \beta < 1$;
- (ii). as auto-correlações decrescem hiperbolicamente em vez de exponencialmente, implicando em uma função de auto-correlação não totalizável $\sum_k r(k) = \infty$, dependência de longa-duração (LRD – *Long Range Dependence*), e
- (iii). a densidade espectral (Transformada de Fourier) $f(\cdot)$ perto da origem comporta-se como $f(\lambda) \sim a\lambda^{-\gamma}$, para $\lambda \rightarrow 0$, $0 < \gamma < 1$ e $\gamma = 1 - \beta$.

Os processos auto-similares se contrapõem-se a modelos de tráfego atualmente vigentes na literatura devido a estes últimos serem caracterizados por:

- (i). a variância da média amostral decresce com a variância do tamanho da amostra, i.e., $var(X^{(m)}) \sim am^{-1}$, com $m \rightarrow \infty$;
- (ii). uma função de auto-correlação que decresce exponencialmente (i.e., $r(k) \sim \rho^k$, $0 < \rho < 1$), implicando uma função de auto-correlação totalizável $\sum_k r(k) < \infty$, dependência de curta-duração (SRD – *Short Range Dependence*), e
- (iii). a densidade espectral é limitada na origem.

4. Cenário de Experimentação

Na realização deste trabalho foram utilizadas três ferramentas: um gerador de tráfego auto-similar [11]; três estimadores estatísticos de parâmetro H [8] e um simulador de Redes Neurais (SNNS) [2].

Basicamente, a idéia foi de gerar *traces*¹ com certos valores de parâmetro H . Neste trabalho, utilizou-se um algoritmo proposto originalmente por Mandelbrot [12] e aperfeiçoado posteriormente por Chi [11]. Este algoritmo gera amostras de ruído Gaussiano fracional (*fractional Gaussian noise*, ou fGn) de tempo discreto. Os tráfegos gerados serviram tanto para treinar a rede neural proposta para a estimativa do parâmetro de Hurst como para alimentar os estimadores estatísticos. Os estimadores estatísticos foram usados para verificação da precisão da rede neural.

A metodologia de trabalho utilizada consistiu de simulações e posterior verificação da precisão da proposta neural através da comparação com as aferições dos três estimadores estatísticos. A Figura 4 ilustra bem a metodologia deste trabalho. Os *traces* constituíram um repositório de informações usadas para as estimativas estatísticas e para a estimativa através da rede neural. Os três estimadores estatísticos usaram todo o *trace* (10000 rajadas), ao passo que o estimador neural utilizou apenas 2000 rajadas para $H=0,5$ e 1000 rajadas para o restante. De posse dos resultados de simulação, as comparações de erros foram feitas e apreciadas.

¹ *Traces* são arquivos de dados gerados por *software* que monitoram o tráfego de uma rede de computadores.

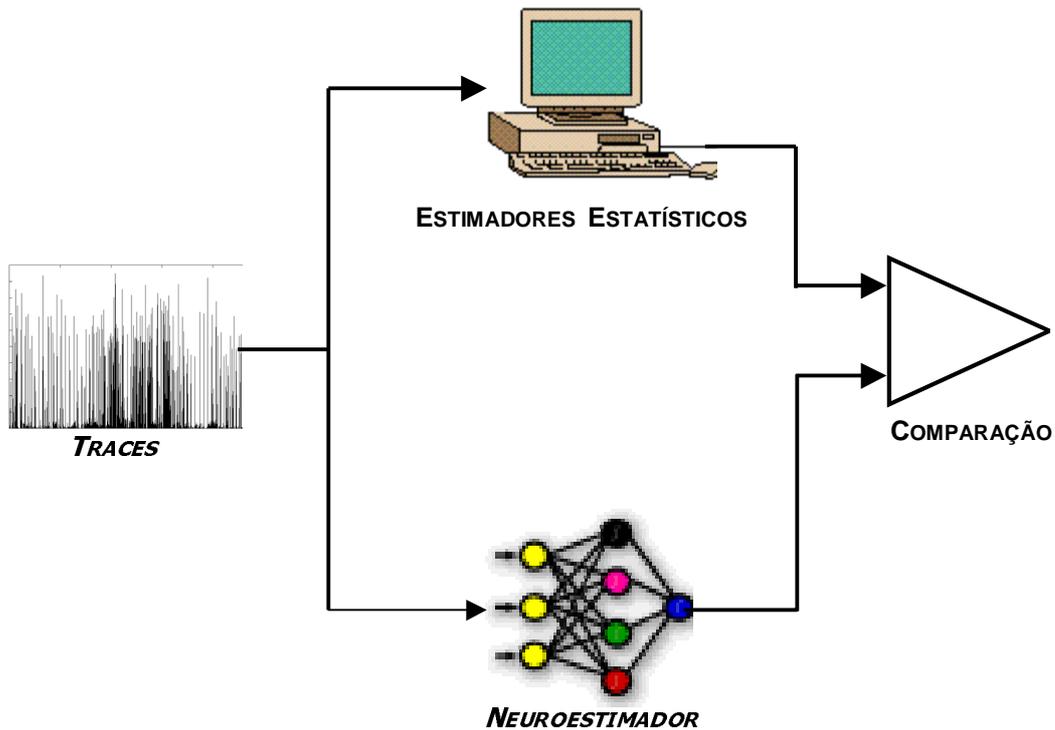


Figura 4: metodologia aplicada.

5. Estimadores

São três os estimadores usados neste trabalho: a estatística R/S (*ReScaled adjusted range*), o método de Higuchi e o estimador de Abry-Veitch. A seguir, suas respectivas descrições.

5.1 Estatística R/S

A estatística R/S foi definida por H.E.Hurst(1951), um hidrólogo que se dedicou a resolver o problema de projetar um reservatório ideal para regular o fluxo do Nilo, baseando-se em registros de flutuações de nível já observadas. O estimador R/S é um dos mais conhecidos e simples métodos para estimar o grau de dependência de uma seqüência amostral. Para um processo estocástico $x(t)$ definido em intervalos de tempo discretos $\{x_t, t = 0, 1, 2, \dots\}$, a faixa reescalada de $x(t)$ em um intervalo de tempo N é definida como a razão R/S:

$$\frac{R}{S} = \frac{\max_{1 \leq j \leq N} \left[\sum_{k=1}^j (X_k - jM(N)) \right] - \min_{1 \leq j \leq N} \left[\sum_{k=1}^j (X_k - jM(N)) \right]}{\sqrt{\frac{1}{N} \sum_{j=1}^N (X_j - M(N))^2}}, \quad (3)$$

com $M(N)$ sendo a média amostral no período N :

$$M(N) = \frac{1}{N} \sum_{j=1}^N X_j$$

O numerador da razão (3) é a medida da faixa (*range*, denotado R) do processo e o denominador é a variância amostral (*sample*, denotado S). Para um processo auto-similar, a razão (3) tem a seguinte característica para um grande N :

$$R/S \sim (N/2)^H, \text{ com } H > 0,5.$$

que pode ainda ser escrito da seguinte forma:

$$\log[R/S] \cong H \log(N) - H \log(2).$$

Com um gráfico de $\log(R/S)$ versus $\log(N)$, o resultado seria aproximadamente uma linha reta com inclinação H . No nosso caso, o valor do parâmetro H pôde ser estimado através de regressão linear sobre os pontos obtidos pelo estimador R/S .

5.2 Método de Higuchi

Este método, sugerido por Higuchi [13], considera a dimensão fractal [14] D de uma série temporal no cálculo do comprimento de uma curva. O método toma as somas parciais $Y(n) = \sum_{i=1}^n X_i$ de uma seqüência de amostras aleatórias $\{X_i\}, i = 1, \dots, N$ – por exemplo, na composição de um fBm a partir de um fGn – e calcula o tamanho normalizado da curva pela fórmula

$$L(m) = \frac{N-1}{m^3} \cdot \sum_{i=1}^m \left[\frac{N-i}{m} \right]^{-1} \cdot \sum_{k=1}^{\lfloor (N-i)/m \rfloor} |Y(i+km) - Y(i+(k-1)m)|,$$

na qual N é o tamanho da série temporal, m é basicamente o tamanho de um bloco e o operador $\lfloor \cdot \rfloor$ denota o maior inteiro inferior ao operando. Verifica-se que $E[L(m)] \sim C_H m^{-D}$, em que $D = 2-H$. Portanto, semelhante ao caso da estatística R/S , um gráfico de $\log(L(m))$ versus $\log(m)$ deverá produzir uma reta com inclinação $D = 2-H$.

5.3 Estimador Abry-Veitch

O estimador Abry-Veitch decompõe uma seqüência de amostras em coeficientes de aproximação (filtro passa-baixas) e detalhe (filtro passa-altas) utilizando a transformada discreta de *wavelet* (*Discrete Wavelet Transform* – DWT). A transformada de *wavelet* assemelha-se à Transformada de Fourier, mas naquela, um sinal temporal $x(t)$ é transformado não no domínio da frequência mas em um domínio de escala temporal.

Os coeficientes de detalhe e aproximação são obtidos através de filtros digitais especialmente projetados. Partindo da seqüência original de amostras, sucessivas seqüências de aproximação e de detalhe são calculadas. Estas seqüências são obtidas aplicando-se filtragem digital recursiva, ou seja, a saída de um estágio de filtragem é novamente aplicada ao estágio de filtragem e assim por diante.

Uma explanação detalhada deste método pode ser encontrada nos trabalhos de Abry [15] e de Veitch [16]. Verifica-se realmente que os algoritmos deste estimador são bem rápidos, apresentando uma baixa complexidade computacional.

6. Projeto da Rede Neural

O projeto da rede neural segue algumas etapas básicas para sua efetivação. Quanto mais elaboradas forem os procedimentos iniciais, notadamente no pré-processamento, mais provável será o sucesso obtido na solução obtida pela rede neural.

Particularmente, houve um certo cuidado quanto à formação dos padrões de treinamento e de testes dos bancos de exemplos. Os dados de entrada foram normalizados para se evitar problemas com paralisia de aprendizado devido à saturação prematura dos pesos. A rede neural escolhida foi do tipo não-recorrente com algoritmo *backpropagation* com *momentum* : sua confiabilidade já é razão suficiente para a escolha [17]. A quantidade de neurônios nas camadas de entrada e de saída foi definida com base na estrutura do problema. Todavia, para o caso da camada escondida, experimentações foram realizadas com variações no tipo de funções de ativação e no número de neurônios.

Uma das fases mais relevantes do projeto é a definição das variáveis de entrada e saída da rede neural. Para cada problema deve existir um conjunto de valores que represente o mais fielmente possível o ambiente onde a rede está inserida.

É claro que a variável de saída é o parâmetro H , propósito deste trabalho. Para as variáveis de entrada seguimos o seguinte raciocínio: o parâmetro H revela o quão auto-similar é o tráfego. Quanto mais próximo de um for valor de H , mais similar será o tráfego e vice-versa. Logo, o próprio tráfego constitui, naturalmente, o elemento mais conveniente para formar as variáveis de entrada da RNA.

Para compor as variáveis de entrada foram escolhidas seqüências de 10 amostras de tráfego. Abaixo seguem dois fortes motivos, um de ordem teórica e outro prático, justificando essa opção:

- i. a característica da auto-similaridade é a repetição do comportamento em diversas escalas. Com base neste fato, uma seqüência de 10 amostras, teoricamente, tem um perfil semelhante a seqüências de 10^2 , 10^3 ou 10^6 amostras, por exemplo;
- ii. além disso, menos neurônios implica em um menor tempo de treinamento.

Desta maneira, as variáveis para a rede neural estão definidas. Indiretamente, a topologia também está quase toda composta. É sabido que há 10 neurônios na camada de entrada, 1 na camada de saída. Optou-se por usar uma camada escondida. Para as camadas escondidas não há ainda procedimento estabelecido de escolha nem de quantas camadas devem ser usadas nem de quantos neurônios ocultos são necessários. É o método da tentativa e erro. A Seção 7 mostra como o número de neurônios escondidos foi escolhido, com base nas respostas de um treinamento prévio de seleção.

Para os dados de treinamento, selecionamos um banco com 600 padrões. Cada padrão tem 11 elementos: dez para a entrada da RNA e um para a saída desejada. As dez amostras submetidas à camada de entrada da RNA constituem uma seqüência de dez rajadas de *trace*, isto é, cada padrão que alimenta a entrada da rede neural é um vetor cujos elementos são os valores de 10 rajadas seguidas. Os padrões foram criteriosamente escolhidos dentro dos *traces* da seguinte maneira:

- 200 seqüências para o *trace* com $H=0,5$;
- 100 seqüências para o demais *traces* ($H=0,6$; $H=0,7$ $H=0,8$ e $H=0,9$).

Note que o *trace* com $H=0,5$ detém o dobro do número de padrões dos demais. Isto não é por acaso: este é o único valor de H para o tráfego com dependência de curta duração (SRD – *Short Range Dependence*). Um simples desvio no seu valor estimado já é o suficiente para uma classificação errônea da sua dependência temporal.

A rede neural aprende através dos dados de treinamento. Depois desta fase, para verificar realmente a ocorrência do aprendizado, observa-se o seu desempenho perante os padrões de testes, os quais não devem ter sido submetidos à rede; devem ser padrões inéditos. A Figura 5 mostra os bancos de padrões utilizados nas experimentações.

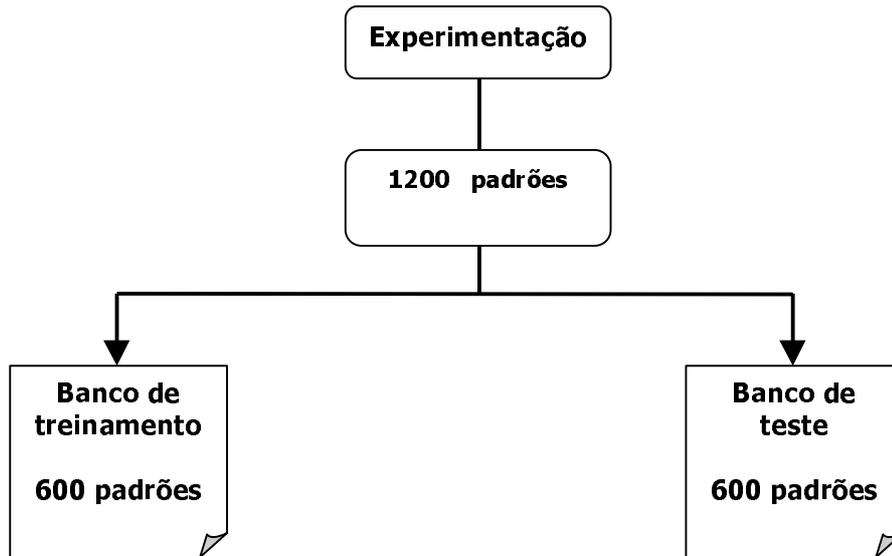


Figura 5: divisão dos bancos de exemplos.

7. Resultados Numéricos

Os gráficos dos arquivos dos *traces* gerados com o parâmetro $H=0,5$ e $H=0,9$ – os casos com os dados gerados de menor e maior auto-similaridade – são mostrados na Figura 6: os *traces* completos são mostrados em (a) e (b), enquanto (c) e (d) contêm as 1000 primeiras rajadas geradas. Observa-se na Figura 6 que as amplitudes das rajadas para o tráfego com $H=0,9$ são maiores em sua maioria que as do tráfego com $H=0,5$, ou seja, realmente o *trace* mais auto-similar é mais “explosivo”

As estimativas e os respectivos erros para o parâmetro H dos três métodos estatísticos estão na Tabela 1.

| Parâmetro H (<i>traces</i>) | Método de Higuchi | | Estatística R/S | | Método de Abry-Veitch | |
|------------------------------------|-------------------|---------|-----------------|----------|-----------------------|-----------|
| | Estimativa | Erro | Estimativa | Erro | Estimativa | Erro |
| 0,5 | 0,4859 | 0,0141 | 0,54378 | -0,04378 | 0,500662 | -0,000662 |
| 0,6 | 0,56521 | 0,03479 | 0,57806 | 0,02194 | 0,591604 | 0,008396 |
| 0,7 | 0,637 | 0,063 | 0,67497 | 0,02503 | 0,691107 | 0,008893 |
| 0,8 | 0,70565 | 0,09435 | 0,76723 | 0,03277 | 0,798722 | 0,001278 |
| 0,9 | 0,78278 | 0,11722 | 0,85429 | 0,04571 | 0,918102 | -0,018102 |

Tabela 1: erro dos estimadores estatísticos.

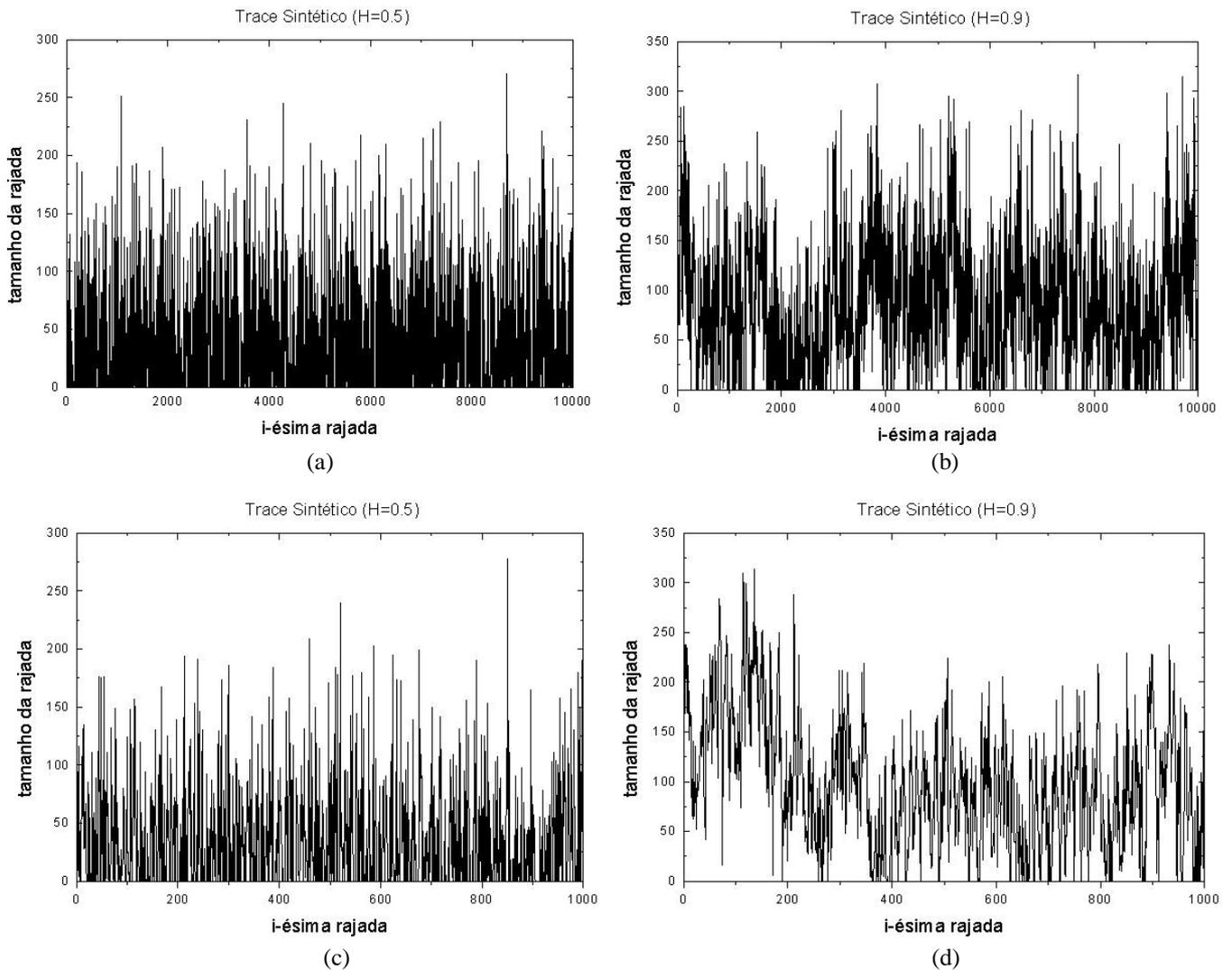


Figura 6: *traces* (10000 amostras): (a) seqüência completa para $H=0,5$; (b) seqüência completa para $H=0,9$; (c) 1000 rajadas iniciais para $H=0,5$; (d) 1000 rajadas iniciais para $H=0,9$

Conforme mencionado na Seção 6 com relação à camada escondida, para a definição do número de neurônios e da função de ativação neles utilizada, efetuou-se um treinamento prévio com 1000 épocas² em variando estes dois elementos.

Portanto, o objetivo deste treinamento inicial não foi o de ensinar a rede mas, antes disso, selecionar a topologia de rede que melhor se adaptasse ao problema. Posteriormente, tomando por base erros médios quadráticos finais (na milésima época) das variações, escolheu-se a configuração de camada escondida que proporcionou o menor erro e aplicou-lhe o método de aprendizado *backpropagation momentum*, treinando-a com um outro critério de parada: um valor pequeno de erro.

A Tabela 2 ilustra as variações testadas. Observa-se logo a diferença de magnitude do erro médio quadrático com relação às duas funções de ativação: a tangente hiperbólica se mostrou muito mais eficiente do que a logística.

² Uma época compreende um ciclo completo (varredura) através de todo o conjunto de padrões de treinamento.

| Número de Neurônios Escondidos | Funções de Ativação da Camada Escondida | |
|--------------------------------|---|-------------------------------------|
| | Logística | Tangente Hiperbólica |
| | Erro Médio Quadrático (1000 épocas) | Erro Médio Quadrático (1000 épocas) |
| 2 | 0.3371 | 0.2191 |
| 5 | 0.3219 | 0.1225 |
| 10 | 0.3190 | 0.0197 |
| 15 | 0.1355 | 0.0060 |
| 20 | 0.2008 | 0.0131 |

Tabela 2 erro médio quadrático do treinamento seletivo em diferentes topologias

Outra observação imediata relevante é que, tanto com a função logística quanto com a tangente hiperbólica, os piores e os melhores casos aconteceram com 2 e 15 neurônios, respectivamente. Uma inspeção na Tabela 2 revela que a rede neural com uma camada intermediária de 15 neurônios e com funções de ativação do tipo tangente hiperbólica é a melhor opção para se começar o treinamento de fato. A Figura 7 ilustra melhor a idéia da Tabela 2.

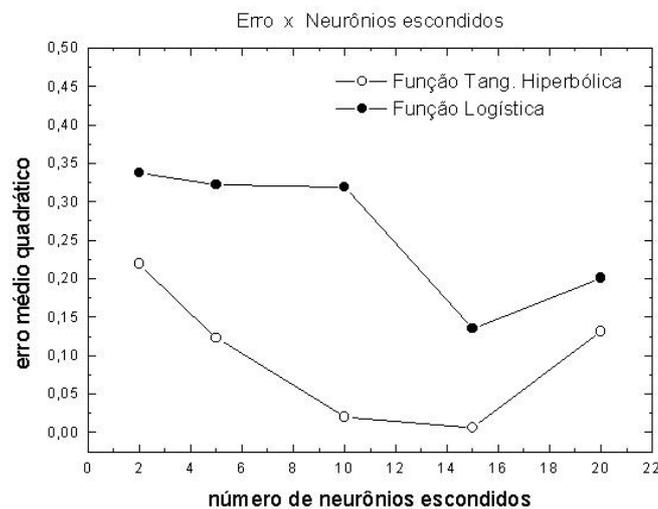


Figura 7: variação dos erros de treinamento em função do número de neurônios e funções de ativação da camada escondida

De posse da estrutura da rede neural escolhida (10 neurônios de entrada, 15 neurônios escondidos e 1 neurônio de saída), o treinamento de fato foi realizado. O critério de parada usado para o aprendizado da rede foi um erro final menor ou igual a 0,0010 (0,1 %). A rede neural foi treinada durante 2616 épocas (vide Figura 8).

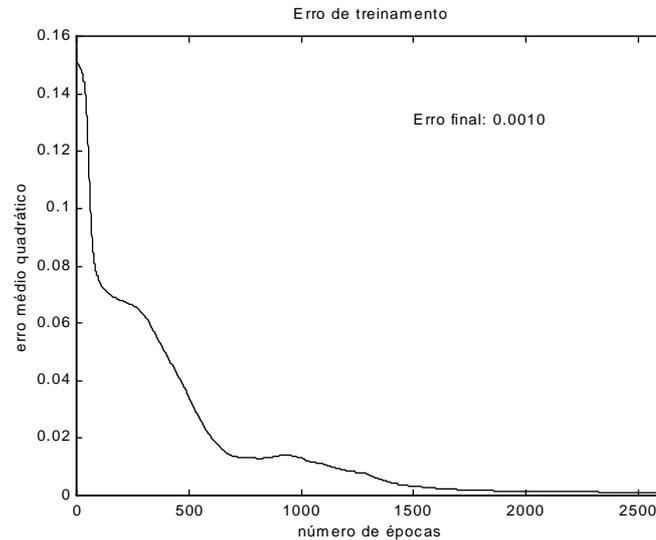


Figura 8: erro de treinamento com a topologia escolhida (10-15-1)

Chegada a fase de execução da rede neural para constatar seu aprendizado, as 600 seqüências do banco de teste (Figura 5) foram aplicadas à RNA. Estas seqüências devem ser inéditas à rede neural, no intuito de verificar sua capacidade de generalização. O erro do estimador neural para cada um dos cinco valores do parâmetro H usados pode ser visto na Figura 9. Note um aspecto curioso. O erro tem um aumento inicial, de $H=0,5$ a $H=0,6$, depois decresce até seu valor mínimo em $H=0,9$. O perfil geral do erro é decrescente à medida que cresce o valor de H , ou seja, a auto-similaridade do tráfego. O valor menor em $H=0,5$, quebrando a regra uniforme decrescente, deve-se ao fato de se ter apresentado à RNA o dobro do número de padrões de treinamento (200) para $H=0,5$.

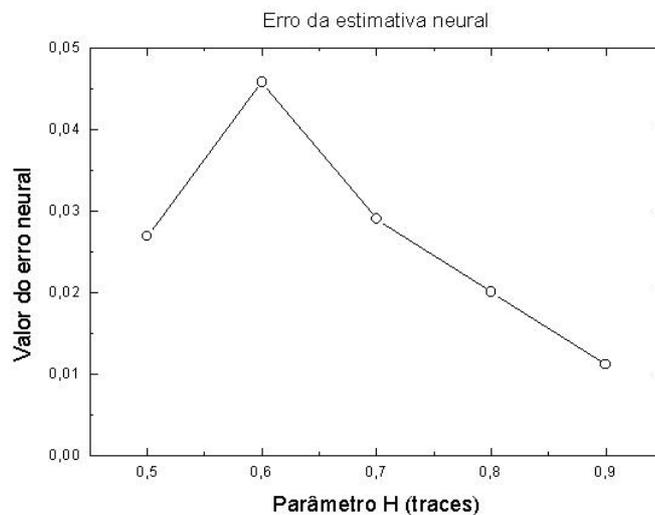


Figura 9: evolução do erro do estimador neural.

Finalmente, com os dados da estimativa neural proposta, pode-se avaliar o grau de precisão da proposta deste trabalho. A Figura 10 exibe as estimativas do parâmetro H de todos os estimadores considerados: os três estatísticos e a proposta neural.

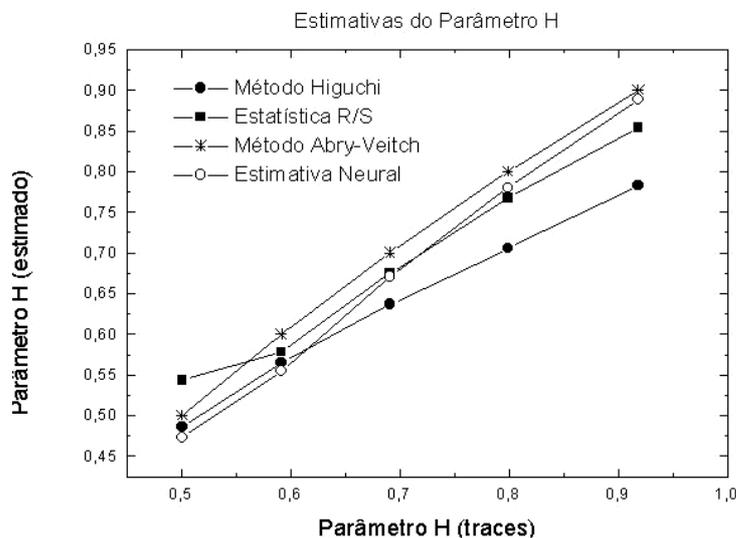


Figura 10: comparação entre os estimadores neural e estatísticos: validação da proposta.

Comparando as estruturas dos estimadores, os estatísticos usam todo o *trace*, ou seja, as 10000 amostras são usadas na estimativa do valor do parâmetro H . A rede neural proposta usou 200 seqüências para o primeiro *trace* e 100 seqüências para os demais. Cada seqüência deteve 10 amostras de tráfego.

O inconveniente da proposta neural é a demora do processo de treinamento. Enquanto os estimadores estatísticos calculam o parâmetro H de uma forma bem direta e rápida, a fase de aprendizado de uma rede neural é bastante dispendiosa. Em contrapartida, a abordagem via redes neurais, dispensa a necessidade de um modelo matemático do problema.

Para uma análise final e comparativa, são necessários os dados da Tabela 3.

| Parâmetro H (traces) | Método de Higuchi | Estatística R/S | Método de Abry-Veitch | Proposta RNA |
|---------------------------|-------------------|-----------------|-----------------------|--------------|
| | Erro | Erro | Erro | Erro |
| 0,5 | 0,0141 | -0,04378 | -0,000662 | 0,0269 |
| 0,6 | 0,03479 | 0,02194 | 0,008396 | 0,0458 |
| 0,7 | 0,063 | 0,02503 | 0,008893 | 0,029 |
| 0,8 | 0,09435 | 0,03277 | 0,001278 | 0,02 |
| 0,9 | 0,11722 | 0,04571 | -0,018102 | 0,0111 |

Tabela 3: comparação dos erros dos estimadores.

A Tabela 3 revela que, dentre os estimadores estatísticos usados, o de Abry-Veitch teve o melhor desempenho. O segundo melhor foi o clássico R/S e o método de Higuchi realmente apresentou os resultados mais pobres.

O estimador neural proposto foi o segundo melhor, ficando aquém somente do estimador de Abry-Veitch. Há de se ressaltar que, todos os três estimadores estatísticos usam o *trace* completo (10000 amostras) para efetuar seus cálculos. A rede neural proposta se utilizou apenas de um décimo do tráfego total, 1000 amostras, para os *traces* de longa duração ($H=0,6$; $H=0,7$; $H=0,8$ e $H=0,9$). Quando o número de amostras para o padrão de entrada foi dobrado, ou seja, 2000 rajadas, notou-se uma melhora abrupta na curva do erro (Figura 9). Isto quer dizer que, mesmo com uma baixa representatividade das seqüências, devido à pouca semelhança do tráfego para $H=0,5$, o desempenho da RNA pôde ser melhorado graças à duplicação da quantidade de padrões que vinham sendo apresentados.

8. Conclusão

Dependências de longa duração têm um impacto significativo no dimensionamento e no gerenciamento de tráfego de redes. O parâmetro de Hurst é o indicador utilizado para se expressar o grau de auto-similaridade, bem como a intensidade das dependências de longa duração. Pequenas variações no parâmetro de Hurst implicam em mudanças significativas no gerenciamento de tráfego. Assim sendo, é de capital importância que se possa avaliar o parâmetro H com precisão e rapidamente. Os estimadores estatísticos demandam uma grande amostragem para se obter um valor preciso do parâmetro de Hurst, o que dificulta a detecção de mudanças do mesmo, em especial em transmissões de dados gerados em tempo real, tais como transmissão ao vivo de aplicações de vídeo.

O presente trabalho investigou a efetividade de se estimar o parâmetro de Hurst através de redes neurais. Pode-se concluir que redes neurais, apesar de demandar um tempo significativo para o seu treinamento, representam uma alternativa para estimação acurada e rápida do parâmetro de Hurst. Assim sendo, recomenda-se a investigação desta metodologia no gerenciamento de tráfego em redes de grande porte com tráfego auto-similar.

A contribuição efetiva deste trabalho reside no fato de que não se encontrou até então, na literatura especializada, nenhum trabalho de estimativa de parâmetro H por meio de redes neurais. A abordagem, portanto, foi uma tentativa bem sucedida e pode ser aproveitada.

Referências

- [1] Tavares, F. *Redes Neurais: Uma introdução à Neurocomputação*, Seminfo 2000, Fortaleza-CE.
- [2] U. of Stuttgart, *SNNS - Stuttgart Neural Network Simulator - User Manual*, Version 4.1, 1995. (<http://www-ra.informatik.uni-tuebingen.de/SNNS/>.)
- [3] Fausset, L. *Fundamentals of Neural Networks* Prentice Hall International, New Jersey, 1994.
- [4] School of Mathematical and Computational Sciences University of St Andrews. *History of Mathematicians* JOC/EFR October 1998.
- [5] Schroeder, M. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. New York: Freeman, 1991.

- [6] Willinger, W., Taqqu, M., *Self-Similar Traffic Modeling for High-Speed Networks*. ConneXions, November, 1994.
- [7] Leland et al., *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*, IEEE/ACM Transactions on Networking, February, 1994.
- [8] Pontes., R. e Coelho, R. *Controle de Admissão de Conexões de Vídeo com Dependência Temporal*. 18º. Simpósio Brasileiro de Redes de Computadores, Belo Horizonte, Brasil, 2000.
- [9] Papoulis, A. *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1991.
- [10] Tsybakov, B. and Nicolas D. *On Self-Similar Traffic in ATM Queues: Definitions, Overflow Probability Bound, and Cell Delay Distribution*. IEEE/ACM Transactions on Networking, 5(3):39-409, June 1997.
- [11] Chi,M., Neal,E. and Young, G.. *Practical Applications of Fractional Brownian Motion and Noise to Synthetic Hydrology*. Water Resources Research, 9:1523-1533, December 1973.
- [12] Mandelbrot, B. and Ness, J.W.. *Fractional brownian motions, fractional noises and applications*. SIAM Review, 10:422-437, October 1968.
- [13] T. Higuchi. *Approach to an irregular time series on the basis of the fractal theory*. Physica D, 31:277-283, 1988.
- [14] M. Barnsley . *The Science of Fractal Images*. Springer-Verlay New York Inc., USA,1988.
- [15] P.Abry and D.Veitch, *Wavelet Analysis of Long-Range Dependence Traffic*. IEEE Transactions on Informations Theory, vol. 44, No. 1, pp. 2-15, 1998.
- [16] D.Veitch and P.Abry , *A Wavelet-Based Joint Estimator of the Parameters of Long-Range Dependence*. IEEE Transactions on Informations, vol. 45, No. 3, pp. 878-897, 1998.
- [17] Haykin, S. *Neural Networks, a comprehensive foundation. Second Edition*. McMaster University, Hamilton, Ontario, Canada. Prentice Hall, 1999.