

Redes neurais em análises de qualidade de água

Daniel Christiano¹; Malva Isabel Medina Hernández²; Pedro L. Christiano³

RESUMO

As redes neurais são ferramentas computacionais que apresentam um amplo leque de aplicações. Neste trabalho procuramos fazer uma breve introdução para explicar seu funcionamento. Como exemplo de aplicação, são usados dados relativos à qualidade da água do rio Jaguaribe, em João Pessoa, PB. Os exemplos apresentados evidenciam a importância da técnica e recomendam sua utilização como previsor a outras áreas de estudo da biologia.

Palavras-chave: redes neurais, rio Jaguaribe, previsores.

Neural networks in analysis of water quality

ABSTRACT

Neural networks are computational tools with a broad range of applications. In this paper a brief introduction shows its functioning and data referring to water quality of the Jaguaribe river in João Pessoa, PB, Brazil are used to show different aspects of its use. The instances considered highlight the importance of the technique and recommend its use for forecasting.

Keywords: Neural networks, rio Jaguaribe, forecasting.

1 INTRODUÇÃO

Um dos objetivos mais importantes da ciência é o de compreender a natureza, para, a partir dessa compreensão, formular previsões sobre futuros acontecimentos. Como um passo intermediário desse processo está a construção de modelos, pois a partir deles que as previsões são realizadas. Diferentemente dos modelos da física, que partem de primeiros princípios e, com base em leis bem estabelecidas, fornecem uma relação causal entre a condição inicial dada e a previsão, em biologia os modelos são em geral estatísticos e nos permitem apenas previsões qualitativas. Eventualmente somos capazes de calcular com que probabilidade um evento pode ocorrer, mas raramente podemos fazer previsões numéricas com precisão razoável. Mesmo quando o fazemos, somos obrigados a recorrer a métodos de ajuste de curvas, em que as distâncias entre os dados experimentais e as curvas com as quais procuramos descrevê-los são minimizadas. Esse tipo de método, apesar de fornecer excelentes resultados em muitos casos, tem a importante limitação de exigir do pesquisador um conhecimento prévio, pelo menos qualitativo, da curva a ser utilizada para os ajustes. Compete ao pesquisador a escolha, com base em outros métodos, que podem incluir até mesmo a intuição, da forma da curva a ser usada no ajuste. A partir daí apenas os parâmetros da curva são ajustados para se determinar aqueles que melhor representam os dados experimentais, para a curva escolhida. Nas situações mais complexas, em que está muito longe de ser evidente que tipo de curva poderia representar os dados com alguma precisão, essa pré-condição é fatal e, em muitos casos, torna inviável a realização de previsões acuradas.

Para contornar essa dificuldade, neste trabalho utilizamos uma metodologia conhecida como redes neurais multicamadas para gerar previsões numéricas sobre a evolução de parâmetros característicos da qualidade da água do Rio Jaguaribe, em João Pessoa, PB. São situações em que claramente os conjuntos de dados não apresentam regularidades a priori. Antes de mais nada, é preciso não se deixar enganar pelo nome atribuído à técnica e esclarecer que trata-se apenas de uma metodologia computacional para o ajuste de

uma curva aos dados experimentais. Ou seja, as redes neurais multicamadas que usaremos neste trabalho são programas computacionais que visam ajustar uma curva a um conjunto de dados experimentais. A diferença é que nesse caso o pesquisador não precisa ter conhecimento prévio de qual é a curva a ser ajustada. O pesquisador pode chutar uma forma muito genérica de curva e, durante o procedimento, a rede cuida de ajustar a forma dessa curva de modo a reduzir os erros nas previsões. Essa etapa do procedimento é conhecida como aprendizagem e nela a rede procura se adequar ao conjunto de dados em questão. Esse trabalho está dividido em uma breve apresentação, discussão geral da técnica a ser utilizada, apresentando seus conceitos mais importantes, os princípios básicos sobre os quais se apóia, uma breve explicação sobre a forma como deve ser utilizada e alguns resultados.

1.1 Redes neurais.

Os estudos de redes neurais, que originalmente visavam simular várias funções cerebrais, tiveram a sua origem com o trabalho de McCulloch e Pitts (1943), que propuseram uma teoria geral de processamento de informação, baseada em uma rede de elementos binários (só podiam assumir dois valores – 0 para desligado e 1 para ligado). Esses elementos são eufemisticamente chamados de “neurônios”, embora sejam muito mais simples que seus equivalentes biológicos.

A partir do trabalho de McCulloch e Pitts (1943), os neurônios passaram a ser vistos como um dispositivo computacional do sistema nervoso, que possuíam várias entradas, onde eram representados os dados iniciais e uma única saída, ou seja, as respostas seriam fornecidas de acordo com os dados de entrada (KOVÁCS, 2006).

As redes neurais de multicamadas são redes nas quais a informação é transmitida em um único sentido, partindo dos neurônios da camada de entrada para os neurônios da camada de saída, passando pelos neurônios das camadas intermediárias, sem retro-alimentação, ou seja, a informação nunca volta de uma camada posterior para as camadas anteriores (ROSEMBLAT, 1962).

Redes neurais multicamadas com alimentação unidirecional são geralmente representadas graficamente usando-se blocos, que são chamados de neurônios, ligados por barras, como se pode ver da figura abaixo.

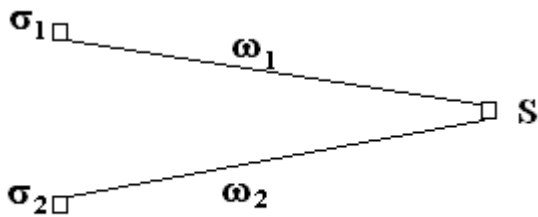


Figura 1 – Rede neural com duas camadas e três neurônios

Nessa figura, que representa a rede neural mais simples possível desse tipo, temos três neurônios (σ_1 , σ_2 e S) e duas camadas. Os neurônios σ_1 e σ_2 compõem a primeira camada, conhecida como camada de entrada e o neurônio S constitui a segunda camada, conhecida como camada de saída. As variáveis ω_1 e ω_2 relacionam os neurônios de entrada aos de saída.

Matematicamente, essa rede representa a relação:

$$S = f\left(\sum_{i=1}^2 \omega_i \sigma_i\right) = f(\omega_1 \sigma_1 + \omega_2 \sigma_2)$$

Onde f é uma função a ser escolhida e é conhecida como função de ativação e cujo papel, comentaremos mais adiante. Dessa forma, o valor atribuído ao neurônio da camada de saída S, é obtido como sendo uma função da soma dos produtos entre os neurônios da camada de entrada σ e os respectivos pesos que os ligam à camada seguinte, ω .

Em 1969, Marvin **Minsky** e **Seymore Papert** mostraram que redes neurais do tipo mostrado figura 1 acima não eram capazes de resolver alguns problemas. Além disso, **Minsky** e **Papert** cometeram um erro que os levou a afirmar que, mesmo com a introdução de mais camadas na rede, esse resultado negativo permaneceria, o que fez com que estudos relativos a esses assuntos fossem abandonados por um bom tempo.

O estudo de redes neurais só voltou a ser novamente um campo de pesquisas ativo em meados da década de 80, quando um método

originalmente proposto por **Werbos** em 1974 para o treinamento de redes neurais com camadas intermediárias, conhecido como algoritmo da retro-propagação do erro ou, como foi originalmente formulado em inglês, error-backpropagation, foi redescoberto e utilizado com sucesso por vários pesquisadores. Logo em seguida esses estudos sofreram um novo grande impulso quando, independentemente, **G. Cybenko** e **K. Hornik**, **M. Stinchcombe** e **H. White** demonstraram, em 1989, que, usando-se redes neurais com uma única camada intermediária, era possível aproximar-se qualquer conjunto de dados. A década de 90 foi então muito rica no estudo de redes neurais e suas aplicações a inúmeros sistemas. Apesar de trabalhos sobre redes neurais, ou que as usam como ferramentas, terem se reduzido a quase zero em revistas de física nos dias de hoje, tais sistemas começam a ser utilizados com ênfase cada vez maior em outras áreas, principalmente na biologia.

Com isso, as redes neurais devem conter pelo menos uma camada intermediária, sendo, de forma geral, representadas como na figura abaixo:

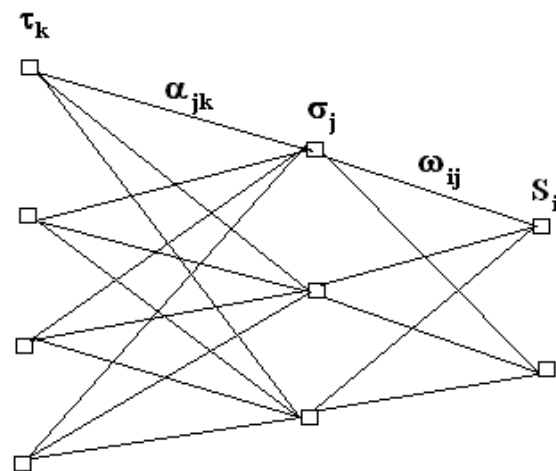


Figura 2 – Rede neural com uma camada intermediária.

Onde os neurônios da camada de entrada são chamados de τ_k , os da camada intermediária de σ_j e os da camada de saída de S_i . As variáveis α_{jk} representam as ligações entre os neurônios das camadas de entrada e intermediária e as variáveis ω_{ij} as ligações entre os neurônios da camada intermediária e os da camada de saída. Matematicamente tem-se que:

$$\sigma_j = f\left(\sum_k \alpha_{jk} \tau_k\right)$$

e

$$S_i = f\left(\sum_j \omega_{ij} \sigma_j\right)$$

de modo que tem-se explicitamente:

$$S_i = f\left(\sum_j \omega_{ij} f\left(\sum_k \alpha_{jk} \tau_k\right)\right)$$

A forma como funciona a rede pode ser melhor compreendida quando se estuda como funciona a etapa conhecida como aprendizagem. É nessa etapa que os parâmetros α_{jk} e ω_{ij} são ajustados para que a rede seja capaz de reproduzir os dados experimentais.

Vamos inicialmente entender melhor o problema que estamos considerando. Os neurônios da camada de entrada representam as variáveis independentes a partir das quais a variável dependente S deve ser obtida. Dessa forma, os neurônios da camada de entrada devem assumir os valores dos dados iniciais e nossa expectativa é a de que S , ao final do cálculo, corresponda à variável que queremos prever. Como não sabemos os valores dos parâmetros α_{jk} e ω_{ij} que devem ser usados para obter a resposta correta, vamos chutar qualquer valor. Da mesma forma faremos para a função $f(x)$. Para essa, entretanto, algumas restrições devem ser impostas, como o fato de ser contínua, com derivada contínua e limitada tanto acima quanto abaixo. Obedecidas todas essas condições, pode ser qualquer função. É evidente que procedendo assim, não vamos achar o valor correto. É daí que começa o processo de aprendizagem. Para tanto usamos um conjunto grande de condições iniciais para quais conhecemos os valores corretos. Para cada grupo desse conjunto nós podemos calcular o valor de S e compará-lo com o valor correto conhecido. A diferença entre os dois será chamada de erro. Nosso objetivo então é alterar os valores dos parâmetros α_{jk} e ω_{ij} de tal maneira que o erro diminua. Isso pode ser feito, como mencionamos anteriormente, usando-se o

algoritmo conhecido como retro-propagação do erro. O nome vem do fato da técnica fazer as alterações dos parâmetros do final para o começo, ou seja, primeiro os pesos ω_{ij} são modificados para depois se proceder à modificação dos pesos α_{jk} .

A idéia por trás desta técnica é muito fácil de ser compreendida. Como queremos apenas compreender a idéia, vamos simplificar e supor que S dependa linearmente de apenas uma variável de entrada τ através de um único peso ω , ou seja:

$$S = \omega \tau$$

Consideremos então um dado momento no treinamento, em que conhecemos tanto o valor de entrada, τ , quanto o valor correto de saída, T . Usando o valor conhecido da variável independente (aquela que é medida) calculamos o valor de S (que, no momento, é o valor previsto para a variável dependente). Como sabemos o valor correto para essa variável, que também foi medido, podemos calcular o erro simplesmente fazendo a diferença entre eles. Por conveniência é melhor usar o quadrado dessa diferença, ou seja:

$$E^2 = (T - S)^2$$

Ou, mais explicitamente:

$$E^2 = (T - \omega \tau)^2 = \omega^2 \tau^2 - 2\omega \tau T + T^2$$

Podemos ver então que o quadrado do erro que cometemos quando calculamos S (nossa previsão) é uma função quadrática do peso ω , que se quer ajustar. Graficamente tem-se

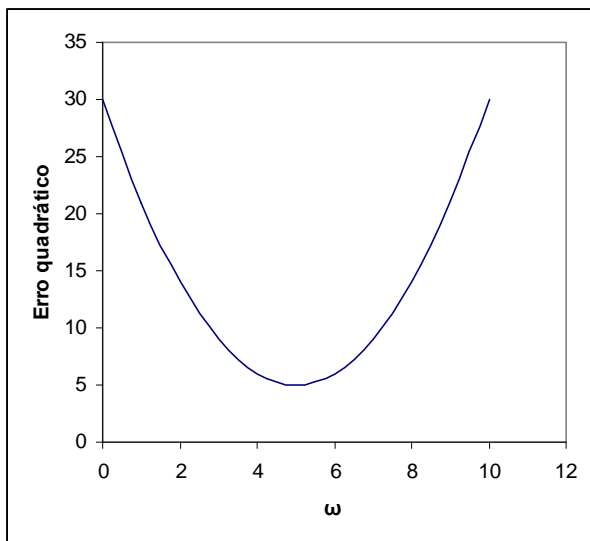


Figura 3 – Erro quadrático em função do peso ω .

Vamos observar esse gráfico. O que observamos em primeiro lugar, é que o valor de ω que minimiza o erro é $\omega = 5$. Dessa forma, se estivermos usando um valor menor de ω desejamos aumentá-lo. Por outro lado, se estamos usando um valor maior desejamos diminuí-lo. Mas como saber se o valor que estamos usando é maior ou menor do que o que minimiza o erro quando não conhecemos esse valor? Basta verificar no gráfico que a derivada da função Erro Quadrático, que pode se facilmente calculada, é negativa quando o valor de ω é menor do que o valor que minimiza essa função e positiva quando ω é maior. Essa observação também nos permite inferir imediatamente uma regra para a correção de ω para aproximá-lo mais do valor que minimiza o Erro Quadrático. Se a derivada for negativa, ω é menor do que o ideal e, para aumentá-lo, deve-se trocar o sinal da derivada e adicioná-la ao valor de ω para obter um novo valor. Como essa derivada pode ser muito grande, essa soma simples poderia levar a um valor de ω muito grande, afastando-o mais do valor desejado. Para evitar isso, faz-se uma correção proporcional à derivada e não exatamente igual à derivada. No caso da derivada ser positiva, exatamente a mesma regra deve ser aplicada, ou seja, troca-se o sinal da derivada e soma-se um valor proporcional a ela ao valor de ω para obter um novo valor de ω . Matematicamente, essa regra pode ser escrita como:

$$\omega_{\text{novo}} = \omega_{\text{velho}} - \mu \frac{dE^2}{d\omega}$$

onde μ é o fator de proporcionalidade a que nos referimos anteriormente. Se esse fator for bem escolhido, o novo valor do parâmetro ω , chamado ω_{novo} , fornecerá um erro menor – e, conseqüentemente uma previsão melhor – do que o valor de ω usado anteriormente – e que, na equação, é chamado de ω_{velho} . Como esse novo valor, podemos repetir todo o procedimento: calcular o erro quadrático, sua derivada e aplicar a mesma relação para achar um novo ω ainda mais próximo do valor ideal e assim até que novas correções não mais reduzam o erro. Nesse momento então teremos chegado, senão ao valor ideal de ω que minimiza o erro, pelo menos ao valor mais próximo dele que podemos obter.

E essa é a idéia básica de uma rede neural. É claro que se as funções são muito complicadas, com muitas variáveis independentes, tem-se que usar redes neurais com vários neurônios de entrada e pelo menos uma camada de neurônios intermediária. Nesse caso então, as relações serão muito mais complicadas do que as que descrevemos aqui. Além disso, não temos nenhuma garantia de que a função erro quadrático seja simples, com um único mínimo.

Por outro lado, existem inúmeras vantagens que justificam o uso do método mesmo nessas condições. Por exemplo, por mais complicadas que sejam as equações que devem ser usadas, não é preciso conhecê-las, pois existem pacotes computacionais com algoritmos já prontos. O que devemos nos preocupar em determinar são as seguintes características da rede:

1. O número de neurônios na camada de entrada. Em geral, isso pode ser determinado das características do problema, mas mesmo nos casos em que isso não é possível, pode-se usar diferentes números testando-se para ver qual fornece o melhor resultado.
2. O número de camadas intermediárias e o número de neurônios em cada uma. Há um compromisso aqui. Mais camadas, com mais neurônios, podem fornecer resultados melhores, mas à custa de um tempo de

computação geralmente muito maior. Alguns testes podem indicar quais os valores mais adequados para essas quantidades.

3. A função de ativação $f(x)$ a ser usada. Os pacotes computacionais disponíveis oferecem várias opções e alguns testes também podem dar indicações sobre a mais adequada.
4. Finalmente, deve ser também previamente determinado o valor da constante de proporcionalidade μ a ser usada. Valores muito grandes podem fornecer resultados desastrosos, afastando o valor de ω do melhor ao invés de aproximar. Valores muito pequenos, apesar de garantirem que se está sempre se aproximando do melhor valor, podem fazer com que essa aproximação seja muito lenta, aumentando exageradamente o tempo de computação. Novamente o procedimento mais simples é o de se testar diferentes valores e verificar os que parecem fornecer os melhores resultados.

No que segue, usamos redes neurais para reproduzir a evolução de parâmetros relativos à qualidade da água do rio Jaguaribe, em João Pessoa.

2 MATERIAIS E MÉTODOS

Os dados utilizados foram fornecidos pela SUDEMA (Superintendência de Administração do Meio Ambiente) e incluíram amostras mensais entre Março de 1998 a Dezembro de 2006. Foram analisados os seguintes fatores: o nível de coliformes fecais, OD, DBO e pH nos quatro pontos de coleta: JB01, JB02, JB03 e JB04.

Os valores fornecidos para alguns parâmetros apresentaram alguns erros, possivelmente de digitação, que foram suprimidos utilizando um método freqüentemente utilizado em estatística, que consiste na eliminação de dados considerados “outliers”, ou seja, dados com valores acima ou abaixo da média $\pm 3,9$ desvios-padrão.

Após a coleta indireta dos dados (SUDEMA) e análise dos mesmos, foi construída uma rede neural, que é um previsor

do tipo multicamadas com alimentação unidirecional da entrada para saída, usando-se o pacote computacional MATLAB.

Os dados disponíveis foram subdivididos em dois conjuntos. O primeiro deles, contendo 72 dados mensais de janeiro de 1998 a dezembro de 2004, foi usado para o treinamento da rede. O segundo conjunto de dados, contendo 24 dados mensais de janeiro de 2005 a dezembro de 2006 foi usado para testar a adequação da rede construída, e esses valores previstos, chamados genericamente de simulação, foram comparados com os valores reais contidos no segundo conjunto de dados.

Foi estabelecido um erro mínimo de 10^{-5} , ou seja, quando o erro atingia esse valor o resultado era considerado satisfatório e o cálculo interrompido, não sendo necessário buscar um erro cada vez menor.

Para a determinação da arquitetura da rede a ser usada em cada caso, foram feitos testes variando a quantidade de neurônios na camada de entrada (NCE) e de neurônios na camada intermediária (NCI). O número de neurônios na camada de saída (NCS), foi fixado em 1.

Na fase de treinamento das redes, testou-se a dependência dos resultados finais com a escolha do parâmetro μ e com a função de ativação utilizada. Foram feitos testes usando-se as funções tangente sigmodal (tansig) e logaritmo sigmodal (Logsig). Para a determinação do valor do neurônio de saída a partir dos da camada intermediária, foi sempre usada a função de ativação purelin, para não limitar os valores de saída – foram testados também os treinamentos: sem e com momento. No primeiro caso, o parâmetro μ é mantido fixo no valor escolhido inicialmente; no segundo caso, esse valor é alterado pelo próprio processo de aprendizagem.

Após esse treinamento foram selecionadas as arquiteturas e funções de ativação que apresentaram os três menores erros finais de aprendizagem, para testar com os dados do segundo conjunto, pois nem sempre um menor erro durante a aprendizagem corresponde a uma melhor simulação, devido a um efeito conhecido como “overtraining”.

Para reduzir as amplitudes de oscilação a intervalos mais fáceis de serem graficados, todas as variáveis foram modificadas

transformando seus valores de forma que ficassem no intervalo entre -1 e +1. Para isso, foi utilizada a transformação:

$$vm = 2*(v-minv)/(maxv-minv) - 1$$

onde:

vm = valor modificado

v = valor bruto

minv = menor valor da série temporal

maxv = maior valor da série temporal.

3 RESULTADOS E DISCUSSÃO

É importante salientar que as redes neurais construídas nesse trabalho apresentam algumas limitações que se devem ao fato da série temporal ser muito pequena, de modo que flutuações atípicas no treinamento ou nos dados de teste podem impedir a geração de uma simulação de qualidade, ou seja, gerando simulações que não conseguem reproduzir adequadamente os dados do conjunto de teste. Isto pode acontecer principalmente por causa da quantidade de flutuações presentes nos dados. Deve-se notar que essas flutuações, por sua própria natureza, são imprevisíveis, de modo que não podem de forma alguma ser previstas pela rede. Também é importante observar-se que os dados usados para treinar a rede continham muitas flutuações, como se pode ver nas figuras 8, 9, 10 e 11 abaixo. Também essa situação é responsável por discordâncias entre os valores previstos pela rede e os realmente medidos. Evidentemente podem existir outras fontes de erros, que não fomos capazes de determinar, mas acreditamos ser essa a que produziu os efeitos mais importantes nos resultados obtidos.

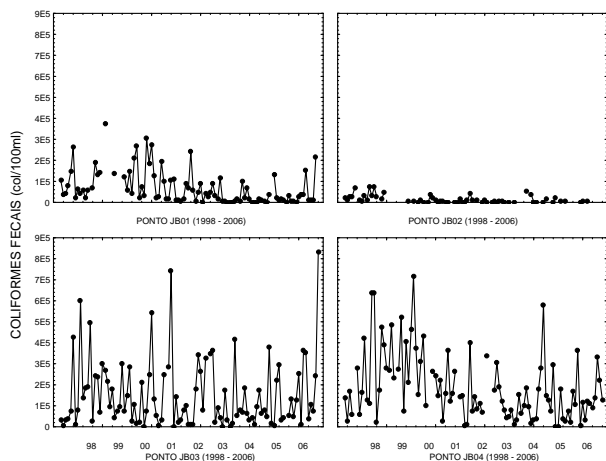


Figura 8: Dados mensais de 1998 a 2006 do nível de coliformes fecais nos 4 pontos de coleta.

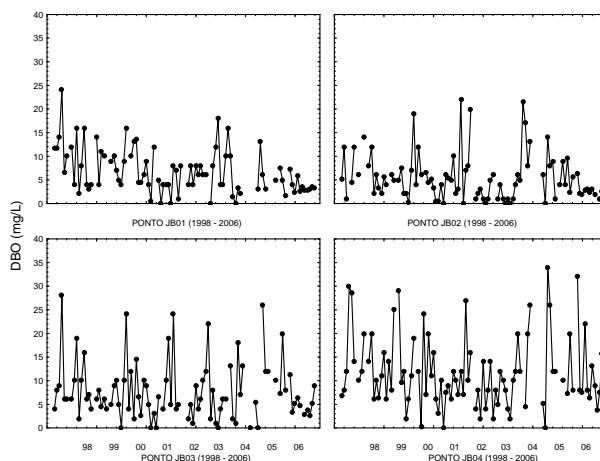


Figura 9: Dados mensais de 1998 a 2006 do nível de DBO nos 4 pontos de coleta.

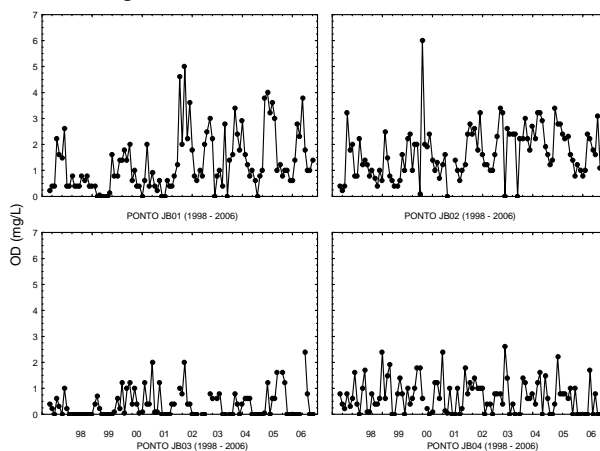


Figura 10: Dados mensais de 1998 a 2006 do nível de OD nos 4 pontos de coleta.

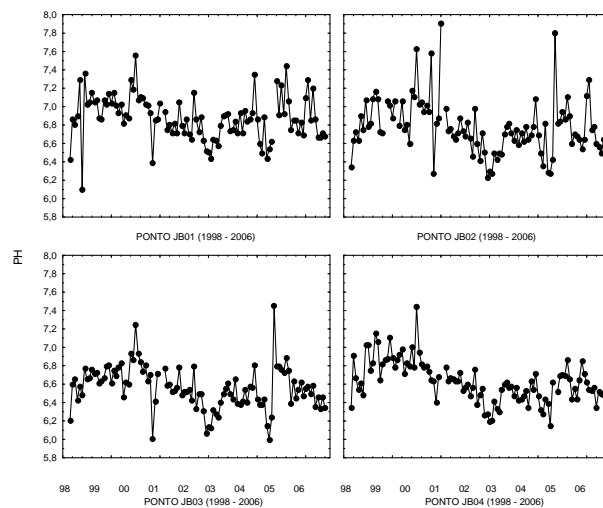


Figura 11: Dados mensais de 1998 a 2006 do nível de pH nos 4 pontos de coleta.

Nas figuras a seguir, os pontos representam os dados reais medidos e as curvas contínuas as previsões obtidas usando-se as redes neurais.

Na figura 12 apresentamos essa comparação usando-se medidas da variável DBO realizadas no ponto 1. Como se pode ver dessa figura, a rede reproduz com bastante precisão o comportamento real, evidenciando a utilidade desta técnica neste tipo de estudo.

Nesse caso foi usada uma rede neural com 3 camadas, com neurônios na camada de entrada, 10 na camada intermediária e 1 na camada de saída. As figuras 13, 14 e 15 se referem a esse mesmo estudo efetuado para outros pontos de coleta no Rio Jaguaribe.

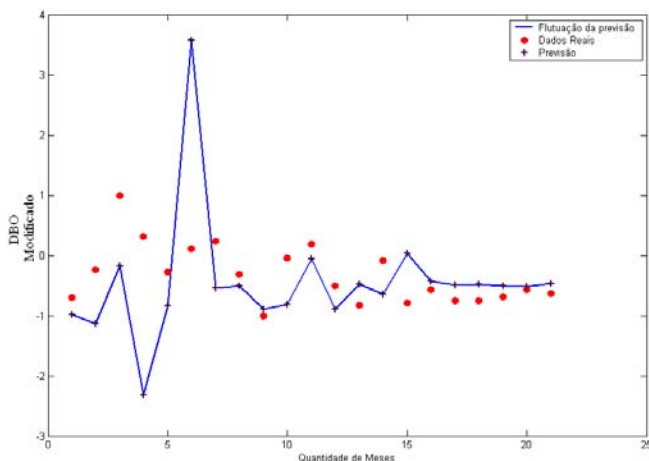


Figura 12: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável DBO no ponto JB01.

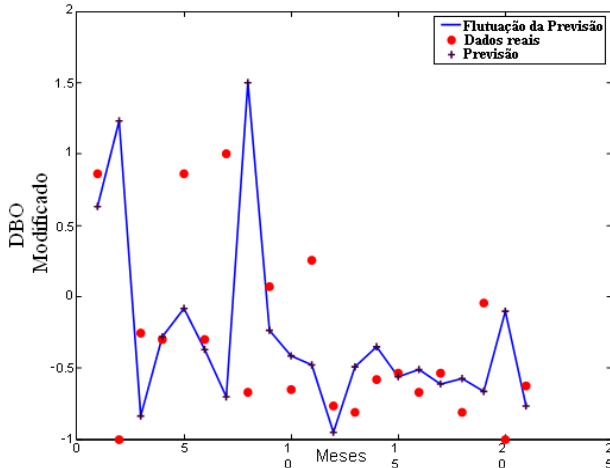


Figura 13: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável DBO no ponto JB02.

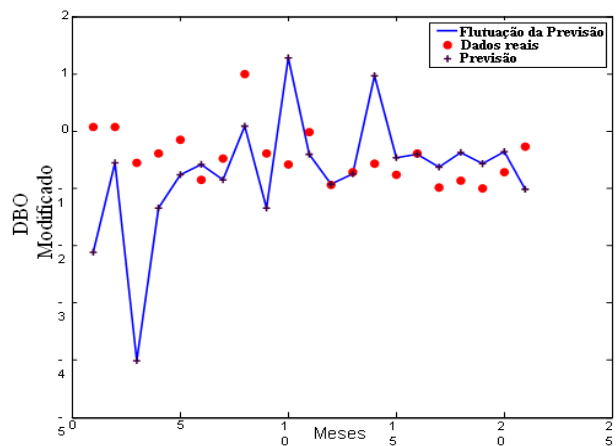


Figura 14: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável DBO no ponto JB03.

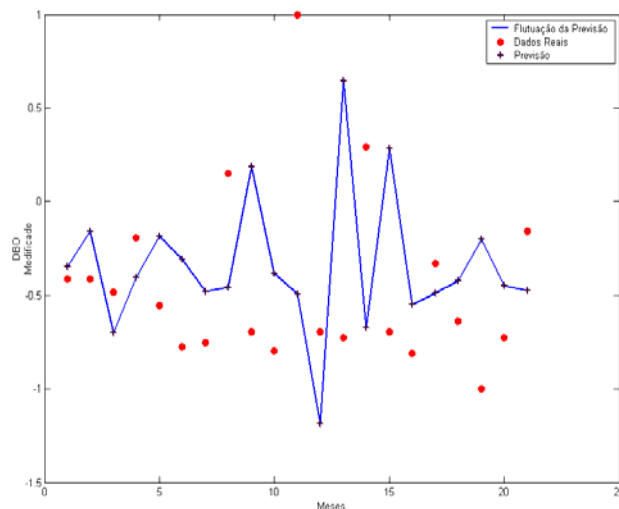


Figura 15: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável DBO no ponto JB04.

As figuras 16, 17, 18 e 19 se referem ao estudo efetuado usando-se dados relativos ao pH da água em 4 pontos diferentes de coleta no Rio Jaguaribe. Nesse caso a rede neural utilizou-se uma rede neural com a mesma arquitetura da anterior. Apenas para o ponto JB03 foi necessária a utilização de um número muito maior de neurônios na camada intermediária (foram usados 30 neurônios) e, mesmo assim, previsões espúrias se mantiveram. Isso ocorreu porque o conjunto de dados usados para o treinamento apresentava flutuações muito maiores do que as apresentadas no conjunto usado para teste. Possivelmente essas flutuações podem ser devidas a erros de mensuração ou de digitação que, apesar de grandes, não atendiam ao critério utilizado para sua eliminação. Isso faz com que a rede “aprenda” a incluir flutuações e,

eventualmente, essas serão incluídas onde não existem na realidade.

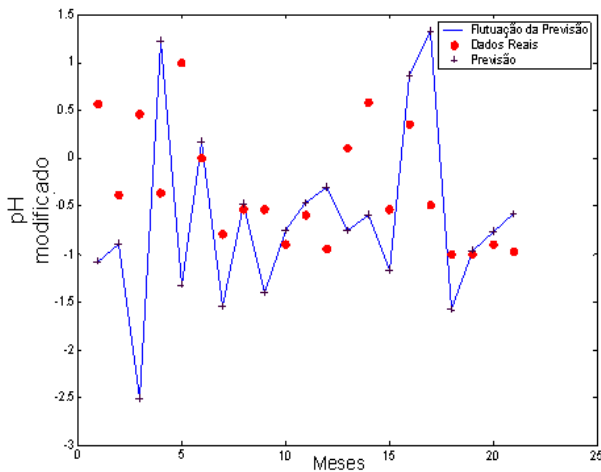


Figura 16: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável pH no ponto JB01.

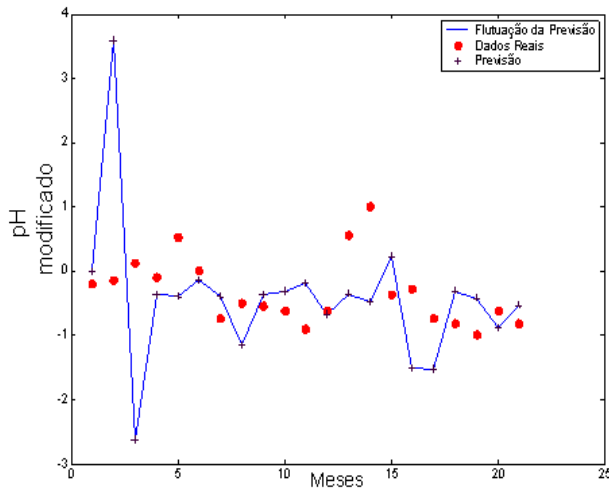


Figura 17: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável pH no ponto JB02.

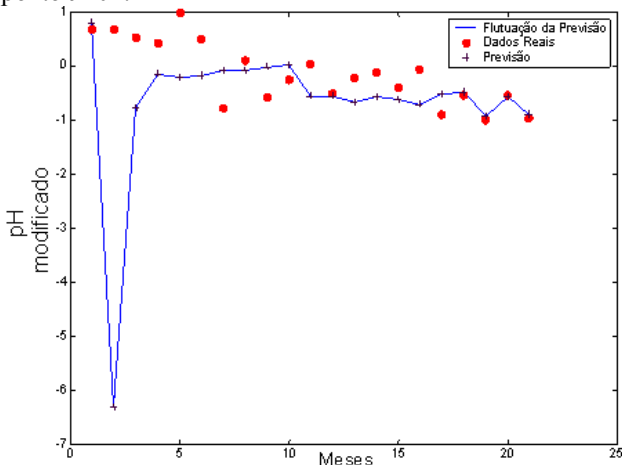


Figura 18: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável pH no ponto JB03.

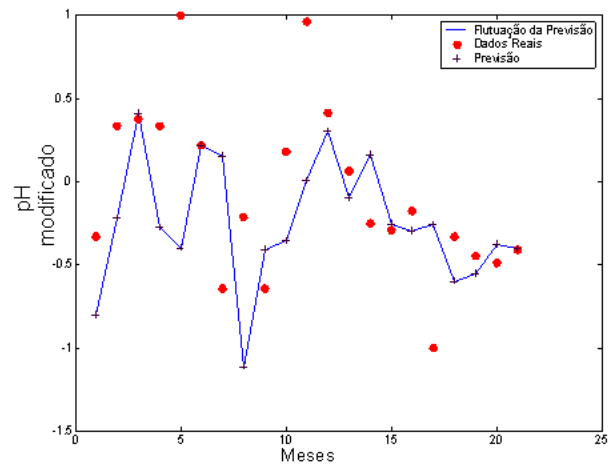


Figura 19: Comparação entre os dados reais, a flutuação da previsão e os pontos da previsão para a variável pH no ponto JB04

4 CONCLUSÕES

Das figuras apresentadas fica claro que se pode fazer previsões quantitativas usando-se redes neurais, mesmo quando se conhece muito pouco sobre o problema que está sendo estudado.

Os resultados também mostram que a qualidade das medições influencia decisivamente na capacidade da rede fazer previsões precisas. Quanto maior a qualidade dessas medições, mais precisas podem ser as previsões realizadas por essa técnica.

Dados de boa qualidade são necessários, para que se tenha certeza que eles apresentam a realidade da forma mais fiel possível. Sem dados realmente confiáveis, a pesquisa pode ser direcionada para um ponto que não seja realmente o principal problema a ser sanado ou minimizado.

Desta forma, as redes neurais construídas provaram poder servir como uma ferramenta, que em conjunto com um monitoramento ambiental cuidadoso, ajude no gerenciamento do ambiente, permitindo previsões precisas e, conseqüentemente habilitando o pesquisador a intervir de forma adequada com maior antecipação.

REFERÊNCIAS BIBLIOGRÁFICAS

CYBENKO, G., Approximation by superposition of a sigmoidal function.

Mathematics of Control, Signals and Systems, v. 2, 303-314, 1989

HORNIK, K.; STINCHCOMBE, M.; WHITE, M., Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, 359-366, 1989.

KOVÁCS, Z. L. *Redes Neurais Artificiais: Fundamentos e Aplicações*. 4ª ed. São Paulo: Editora Livraria da Física, 2006. 171 p.

MINSKY, M; PAPERT, S., *Perceptrons: an introduction to computational geometry*. Cambridge: MIT Press, 1969.

WERBOS, P., *Beyond regression: new tools for prediction and analysis in behavioral sciences*. 1974. PhD. Thesis, Harvard University, 1974.

1 - Mestre em Meio Ambiente e Desenvolvimento pelo PRODEMA

2 - Departamento de Ecologia e Zoologia, Universidade Federal de Santa Catarina, Caixa Postal 476, Florianópolis, SC, Brasil. CEP: 88010-970

3 - Departamento de Física, Universidade Federal da Paraíba, Caixa Postal 5075, João Pessoa, PB, Brasil, Cep: 58051-900 ; pedro@fisica.ufpb.br