

MODELAGEM PREDITIVA DE LINHA DE COSTA UTILIZANDO REDES NEURAS ARTIFICIAIS

Shoreline predictive modeling using artificial neural networks

RODRIGO MIKOSZ GONÇALVES¹
LEANDRO DOS SANTOS COELHO²
CLÁUDIA PEREIRA KRUEGER³
BERNHARD HECK⁴

¹Universidade Federal de Pernambuco (UFPE)
Centro de Tecnologia e Geociências – CTG, Departamento de Engenharia
Cartográfica, Recife-PE, Brasil, rodrigo.mikosz@gmail.com

^{1,3}Universidade Federal do Paraná (UFPR)
Programa de Pós-Graduação em Ciências Geodésicas, Curitiba-PR, Brasil,
ckrueger@ufpr.br

²Pontifícia Universidade Católica do Paraná, (PUC-PR)
Programa de Pós-Graduação em Engenharia de Produção e Sistemas, Curitiba-PR,
Brasil, leandro.coelho@pucpr.br

⁴Karlsruhe Institute of Technology
Geodetic Institute Karlsruhe, Karlsruhe, Alemanha, heck@gik.uni-karlsruhe.de

RESUMO

Estudar modelagens através de dados geodésicos temporais com a possibilidade de prever a posição de linha de costa é uma tarefa importante e pode auxiliar significativamente na gestão costeira. A área de estudo neste trabalho se refere ao município de Matinhos no estado do Paraná, Brasil. As linhas de costa temporais utilizadas para testar a modelagem preditiva são provenientes respectivamente da fotogrametria analógica para anos 1954, 1963, 1980, 1991 e 1997 e de levantamentos geodésicos utilizando GPS (*Global Position System*) para 2001, 2002, 2005 e 2008 (como controle). Dois testes com as redes neurais artificiais foram organizados mudando alguns parâmetros como: arquitetura, número de neurônios nas camadas ocultas e algoritmos de treinamentos. Quando comparados o valor dos resíduos entre a predição e a linha de costa de controle, os melhores resultados estatísticos indicam que o MAPE (*mean absolute percentage error*) são 0,28% utilizando a rede neural parcialmente recorrente de Elman com o algoritmo de treinamento quase-Newton e 0,46% para o caso da rede neural perceptron multicamadas com o algoritmo de treinamento utilizando o método Bayesiano com regularização.

Palavras-chaves: Mapeamento Costeiro; Modelagem Preditiva; Linha de Costa; Redes Neurais Artificiais.

ABSTRACT

The study of models using geodetic temporal data which can possibly predict the shoreline position is an important task and can significantly contribute to coastal management. The studied area is located at municipality of Matinhos in the Paraná State, Brazil. The temporal shoreline used to test the prediction model is respectively from analog photogrammetric data, related to the years 1954, 1963, 1980, 1991 and 1997, and GPS (Global Position System) geodetic surveys for 2001, 2002, 2005 and 2008 (as control). Two different tests with artificial neural network were organized setting the parameters like: architecture, number of neuron in hidden layers and the training algorithms. Comparing the residuals between the prediction to the shoreline of control, the best statistical results show the MAPE (Mean Absolute Percentage Error) is 0,28% using the Elman partially recurrent network with quasi-Newton training function and 0,46% using the neural network multilayer perceptron with Bayesian regulation training function.

Keywords: Coastal Mapping; Artificial Neural Network; Prediction Models; Shoreline.

1. INTRODUÇÃO

As ciências geodésicas têm um papel fundamental para mapear ambientes costeiros e através de mapeamentos históricos de linha de costa é possível verificar a dinâmica de um local em estudo.

O problema de erosão costeira é mundial. No Brasil, a ocupação desordenada vem agravando essa situação e centenas de praias estão sob problemas severos de erosão (ver SOUZA, 2009).

Internacionalmente o modelo preditivo mais aceito para movimentação de linha de costa é o de regressão linear, porém, para o estudo de ambientes costeiros é necessário fazer uso de dados e informações referentes à posição de linha de costa no passado, por exemplo, de 50, 60 anos atrás ou mais e estes dados dependem da qualidade da sua interpretação e extração. Na prática, quando dados antigos são agregados ao conjunto de observações temporais e não são tomados os devidos cuidados estes podem contribuir para representar predições fora da realidade.

Destaca-se que a hipótese de descartar dados antigos não é atraente para o ambiente costeiro, pois os dados antigos são de extrema importância para o resgate de informações. Justifica-se então trabalhar com todos os dados históricos da posição de linha de costa disponíveis, porém com modelos capazes de ponderar os dados antigos de forma eficaz.

A capacidade de uma rede neural artificial (RNA) em aprender através de um conjunto reduzido de exemplos, mapeando relações de entrada e saída e posteriormente dando respostas para dados não-conhecidos, demonstra o poder das

RNAs como ferramenta computacional para solução de problemas complexos, por exemplo, robôs que desarmam bombas e mercado financeiro. Este atrativo é um diferencial em relação à computação convencional que necessita de algoritmos com regras rígidas e pré-fixadas (KRÖSE e SMAGT, 1996, HAYKIN, 1999, BRAGA *et al.*, 2000).

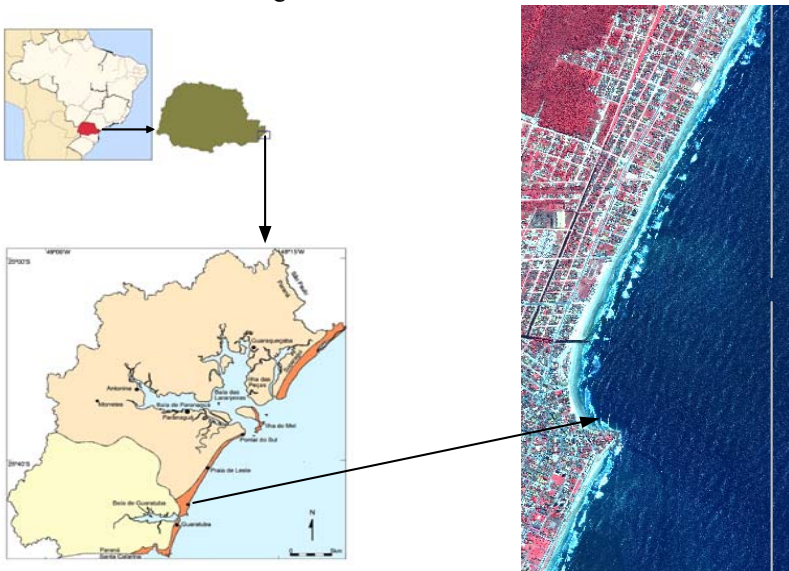
Várias são as aplicações para RNAs, entre elas encontram-se a utilização para: reconhecimento de padrões (WALDEMARK *et al.*, 1995), agrupamento de dados (GUERRA, 2006) processamento de sinais (FUJII, 1993), telecomunicações (GEROMEL, 2003), robótica (KUBOTA *et al.*, 2003), tomada de decisões (MACIEL, 2005), predição (HAYATI e SHIRVANY, 2007) e assim como aplicações na área de ciências geodésicas (WANDRESEN, 2004, SOUZA, 2006, VAFEIDIS *et al.*, 2007).

Detalhes da elaboração do modelo preditivo utilizando RNA, testes realizados e resultados experimentais encontrados com dados do município de Matinhos localizado na costa do estado do Paraná, Brasil, são apresentados neste trabalho.

2. ÁREA DE ESTUDO E FONTE DE DADOS

A área de estudo refere-se a uma região costeira do litoral paranaense de aproximadamente 6 km de linha de costa no município de Matinhos (Figura 1).

Figura 1 – Área de estudo.



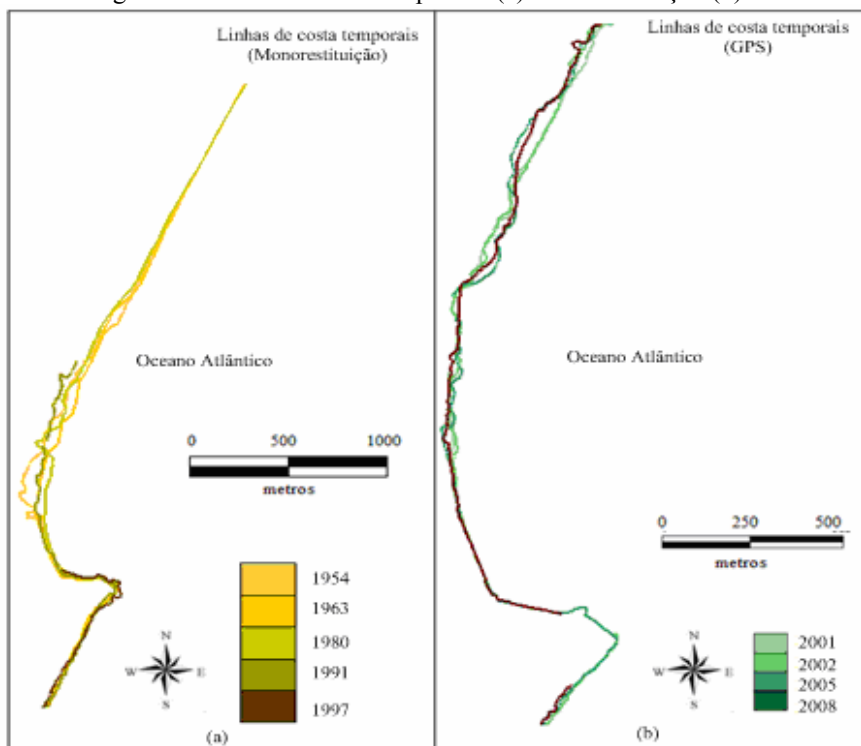
A primeira etapa para predição da linha de costa exigiu esforços para resgatar e organizar os dados temporais em um ambiente chamado de Sistemas de

Informações Geográficas (SIG) Costeiro de Matinhos-PR, (ver, GONÇALVES *et al.*, 2008).

As linhas de costa temporais são os resultados de alguns trabalhos e projetos de pesquisas, entre eles cita-se o projeto DIAMAT (2000). Nele encontram-se as linhas de costa obtidas pela monorestituição digital (ver, MITISHITA, 1997) e dados referentes ao posicionamento relativo cinemático com receptores GPS (ver, KRUEGER *et al.*, 2009).

Na Figura 2 (a) apresentam-se as linhas de costa temporais dos anos de 1954, 1963, 1980, 1991 e 1997 obtidas pela monorestituição digital. Na Figura 2 (b) apresentam-se as linhas de costa obtidas por posicionamentos relativos cinemáticos com o uso de receptores GPS para os anos de 2001, 2002, 2005 e 2008.

Figura 2 - Linhas de costa temporais: (a) monorestituição (b) GPS.



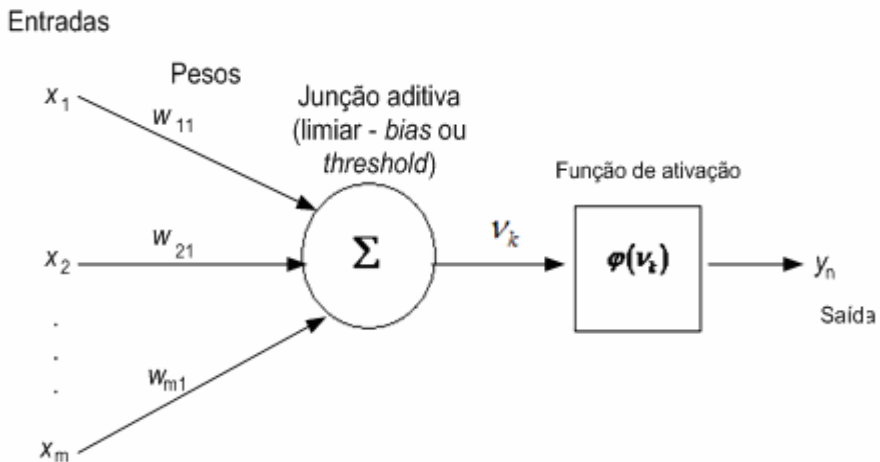
3. REDES NEURAIS ARTIFICIAIS

Para configuração e elaboração de RNAs devem ser especificadas: as características dos neurônios, a topologia e as regras de treinamento. A adaptação

dos pesos iniciais e a aprendizagem do seu comportamento são especificadas pelas regras de treinamento. Os algoritmos de treinamento (ou aprendizado) de uma RNA têm a característica de ajustar de forma iterativa os pesos das conexões entre os neurônios até que os pares de entradas e saídas esperados, no caso de aprendizado supervisionado, por exemplo, sejam obtidos e as relações matemáticas de causa e efeito possam ser estabelecidas. Quando a configuração de um determinado problema apresentado às RNAs muda e o modelo do desempenho não é mais adequado à situação é possível submeter a RNA a mais treinamentos com novas condições de entrada e saída para a melhora do seu desempenho (KRÖSE e SMAGT, 1996; HAYKIN, 1999). Existem também as redes neurais sem peso, onde a principal diferença entre as RNAs convencionais é a forma de armazenar informação (ver, BRAGA *et al.*, 2000).

O primeiro modelo artificial de um neurônio biológico foi proposto no trabalho pioneiro de Warren McCulloch e Walter Pitts em 1943. O modelo esquemático de um neurônio artificial proposto por McCulloch-Pitts é uma simplificação do que se sabia, até então, a respeito do neurônio biológico. A Figura 3 apresenta o modelo geral de um neurônio artificial e nela é possível identificar três elementos básicos (ver, por exemplo em: KRÖSE e SMAGT, 1996; HAYKIN, 1999; ARBIB, 2003; BRAGA *et al.*, 2000).

Figura 3 – Modelo geral de um neurônio artificial.



Adaptado de HAYKIN (1999).

O primeiro elemento é um conjunto de sinapses ou elos de conexão, cada qual caracterizada por um peso w_{kj} , com $k = 1$ e $j = 1, 2, \dots, m$.

O segundo elemento diz respeito a um somatório para os sinais de entrada, ponderado pelas sinapses do neurônio (combinação linear), tal que

$$v_k = \sum_{j=0}^m w_{kj} x_j. \tag{1}$$

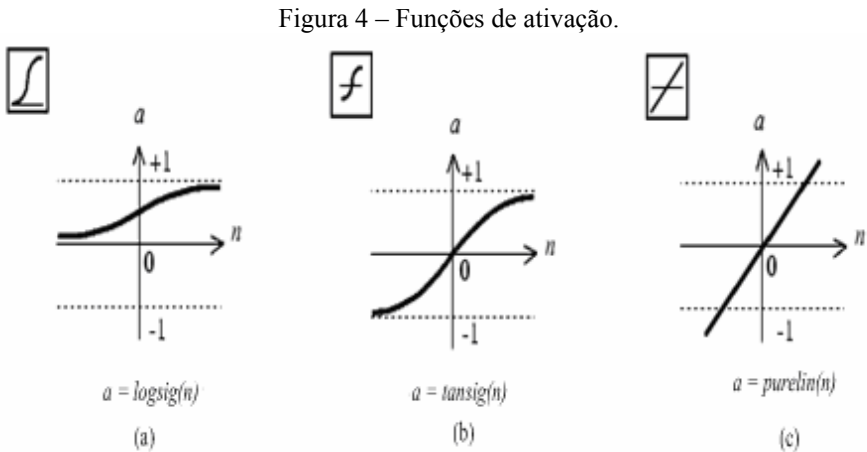
O terceiro elemento é uma função de ativação, que restringe a amplitude de saída de um neurônio, limitando o intervalo de saída para um valor finito

$$y_k = \varphi(v_k) \tag{2}$$

Um neurônio k pode ser descrito matematicamente através das Equações 1 e 2, onde φ representa a função de ativação do neurônio artificial. Esta, existe para buscar um ajuste ótimo ou modelo para um problema.

Vários outros modelos, após o trabalho de McCulloch e Pitts, foram derivados permitindo saídas, não necessariamente iguais a 0 ou 1 e com diferentes funções de ativação.

A Figura 4 apresenta três tipos de funções de ativação. Em (a) a função logística sigmóide (*logsig*); (b) a função tangente hiperbólica (*tansig*); e em (c) a função linear de ativação (*purelin*).



Fonte: NEURAL NETWORK TOOLBOX.

A função logística sigmóide, Equação 3, pode assumir valores entre 0 e 1, onde a é o parâmetro de inclinação da função sigmóide e v é o valor de ativação do neurônio,

$$f(v) = \frac{1}{1 + e^{(-av)}} \tag{3}$$

A Equação 4 representa a função tangente hiperbólica assumindo valores entre 1 e -1, onde a é o parâmetro de inclinação da curva, b são os limites inferiores e superiores e v é o valor de ativação, ou seja,

$$f(v) = a \frac{e^{(bv)} - e^{(-bv)}}{e^{(bv)} + e^{(-bv)}} \quad (4)$$

A função de ativação linear é definida pela Equação 5 onde a é um número real que define a saída linear para os valores de entrada, y é a saída e x é a entrada. Assim,

$$y = ax \quad (5)$$

Um importante conceito de uma RNA é a definição da *arquitetura*, ou seja, a forma com que os neurônios em uma rede podem estar organizados, sendo este um parâmetro importante que restringe o tipo de problema que pode ser tratado na rede. RNAs com uma camada única de neurônios, por exemplo, só conseguem resolver problemas linearmente separáveis. Redes recorrentes, por sua vez, são mais apropriadas para resolver problemas que envolvem processamento temporal (BRAGA *et al.*, 2000).

3.1 Aprendizado

As RNAs têm a capacidade de aprender através de exemplos e fazer interpolações e extrapolações do que aprenderam. Neste tipo de aprendizado conexcionista, não se procura obter regras, mas sim determinar a intensidade de conexões entre neurônios. O conjunto de procedimentos bem-definidos para adaptar os parâmetros de uma RNA para que a mesma possa aprender uma determinada função se chama: algoritmo de aprendizado. Eles basicamente se diferem pela maneira que o ajuste dos pesos é feito, sendo assim existe um conjunto de ferramentas que são representadas por diversos algoritmos, onde cada um tem suas características que podem apresentar vantagens e desvantagens na solução de problemas (BRAGA *et al.*, 2000).

Existem diversos tipos de métodos para treinamento de redes e estes são agrupados em dois tipos: o aprendizado supervisionado e aprendizado não supervisionado. Outros dois paradigmas conhecidos são os de aprendizado por reforço (que é um caso particular de aprendizado supervisionado) e aprendizado por competição (que é um caso particular de aprendizado não supervisionado) (BRAGA *et al.*, 2000).

3.2 Perceptron multicamadas

As redes MLP (*multilayer perceptron*) apresentam um poder computacional muito maior do que as redes sem camadas intermediárias, pois possuem a capacidade de tratar dados que não são linearmente separáveis (HAYKIN, 1999; ARBIB, 2003). Um dos principais aspectos deste tipo de rede esta relacionado com a função de ativação utilizada (BRAGA *et al.*, 2000).

A MLP consiste de um conjunto de unidades sensoriais (neurônios de fonte) que compõe a camada de entrada, uma ou mais camadas ocultas de neurônios computacionais e uma camada de saída de neurônios computacionais. O sinal de entrada se propaga para frente camada por camada através da rede (KRÖSE e SMAGT, 1996).

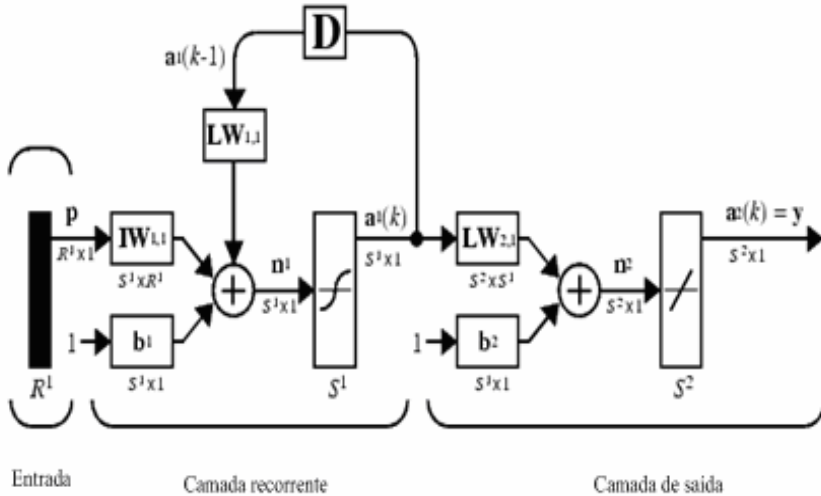
3.3 Redes parcialmente recorrentes de Elman

Nas redes de Elman, além das unidades de entrada, intermediárias e de saída, há também as unidades chamadas de contexto, como no caso das redes parcialmente recorrentes em geral. As unidades de entrada e saída interagem com o ambiente externo, enquanto as unidades intermediárias e de contexto não o fazem. As unidades de entrada são apenas unidades de armazenamento que passam os sinais sem modificá-los. As unidades de saída são unidades lineares que somam os sinais que recebem. As unidades intermediárias podem ter funções de ativação lineares ou não-lineares, e as unidades de contexto são usadas apenas para memorizar as ativações anteriores das unidades intermediárias e podem ser consideradas como atraso no tempo em um passo. As conexões *feedforward* são modificáveis, e as conexões recorrentes são fixas, motivo pelo qual a rede de Elman é apenas parcialmente recorrente (BRAGA *et al.*, 2000).

Em um intervalo de tempo específico k , as ativações das unidades intermediárias (em $k - 1$) e as entradas correntes (em k) são utilizadas como entradas da rede. Em um primeiro estágio, *feedforward*, estas entradas são propagadas para frente para produzir as saídas. Posteriormente, a rede é treinada com o algoritmo de aprendizagem da retropropagação do erro padrão. Após este passo de treinamento, as ativações das unidades intermediárias no tempo k são reintroduzidas, através das ligações recorrentes nas unidades de contexto, sendo salvas nestas unidades para o próximo passo do treinamento ($k + 1$). No início do treinamento, as ativações das unidades intermediárias são desconhecidas e, geralmente são inicializadas para a metade do valor máximo que as unidades intermediárias podem ter (BRAGA *et al.*, 2000).

Na Figura 5 apresenta-se a rede de Elman com duas camadas com realimentação da primeira camada de saída para a primeira camada de entrada.

Figura 5 – Exemplos da arquitetura de Elman.



Fonte: NEURAL NETWORK TOOLBOX

As redes de Elman têm neurônios com a função tangente hiperbólica nas camadas ocultas (recorrentes) e neurônios com a função linear de ativação na camada de saída. Quando a função a ser ajustada aumenta sua complexidade é necessário aumentar o número de neurônios nas camadas ocultas.

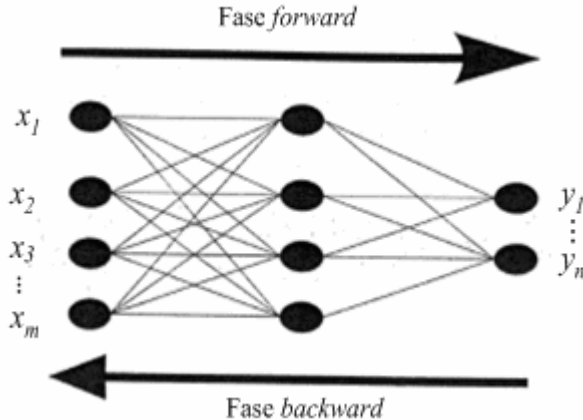
Sua diferença para as redes convencionais está exatamente na conexão recorrente da primeira camada. O atraso nesta conexão armazena valores do passo inicial, que pode ser usado no passo corrente.

3.4 O algoritmo *back-propagation*

O algoritmo mais popular das RNAs é o de retropropagação do erro (*error back-propagation*), cuja fundamentação é corrigir os erros conforme a regra de aprendizagem. A maioria dos métodos de aprendizado para RNAs do tipo MLP utiliza variações deste algoritmo (BRAGA *et al.*, 2000).

Sua aprendizagem consiste em dois passos através das diferentes camadas da rede, conforme o sentido (Figura 6): um passo à frente (*forward*), a propagação, e um passo para trás, a retropropagação (*backward*).

Figura 6 – Fluxo de processamento do algoritmo retropropagação do erro.



Fonte: (BRAGA *et al.*, 2000).

No passo para frente (*forward*) encontram-se os seguintes sub-passos (BRAGA *et. at.*, 2000):

1. A entrada é apresentada à primeira camada da rede, a camada C^0 .
2. Para a camada C^i a partir da camada de entrada, após os neurônios da camada C^i ($i > 0$) calcularem seus sinais de saída, estes servem como entrada para a definição das saídas produzidas pelos neurônios da camada C^{i+1} ;
3. As saídas produzidas pelos nodos da última camada são comparadas às saídas desejadas.

A fase *backward* envolve as seguintes etapas (BRAGA *et al.*, 2000):

1. A partir da última camada, até chegar na camada de entrada:
 - 1.1 Os neurônios da camada atual ajustam seus pesos de forma a reduzir seus os erros nas respostas desejadas dos padrões.
 - 1.2 O erro de um neurônio das camadas intermediárias é calculado utilizando os erros dos neurônios da camada seguinte conectados a ele, ponderados pelos pesos das conexões entre eles.

O algoritmo retropropagação do erro, que faz uso destas duas fases, é apresentado da seguinte maneira:

1. Iniciar os pesos.
2. Existem vários métodos para que o treinamento da rede seja encerrado. Entre os critérios de parada mais utilizados estão:
 - Repetir até a realização de um dado número de épocas (iterações);
 - Repetir até que a diferença entre os erros de duas iterações consecutivas seja próxima de zero ou ficar abaixo de uma constante, ϵ ;

- encerrar o treinamento quando a porcentagem de classificações corretas estiver acima de uma constante (mais indicados para saídas binárias).

2.1 Para cada padrão de treinamento X .

2.1.1 Definir saída da rede através da fase *forward*.

2.1.2 Comparar saídas produzidas com saídas desejadas.

2.1.3 Atualizar pesos dos neurônios através da fase *backward*.

O algoritmo de retropropagação do erro é baseado na regra delta proposto por Windrow e Hoff (1960), sendo por isto também chamada de regra delta generalizado. Este algoritmo propõe uma forma de definir o erro dos nodos das camadas intermediárias, possibilitando o ajuste de seus pesos. Os ajustes dos pesos são realizados utilizando-se o método do gradiente.

A derivação da regra delta generalizada é semelhante à derivação da regra delta. A função custo a ser minimizada é uma função de erro, definida pela soma dos erros quadráticos e representada por

$$E = \frac{1}{2} \sum_p \sum_{i=1}^k (d_i^p - y_i^p)^2 \quad (6)$$

onde: E é a medida de erro total, p é o número de padrões, k é o número de unidades de saída, d_i é a i -ésima saída desejada e y_i é a i -ésima saída gerada pela rede. Esta equação define o erro total cometido pela rede, ou a quantidade em que, para todos os padrões p de um dado conjunto, as saídas geradas pela rede diferem das saídas desejadas.

A regra delta generalizada requer que as funções de ativação utilizadas pelos neurônios sejam contínuas, diferenciáveis e, geralmente, não-decrescentes da entrada total recebida pelo nodo. A Equação 7 ilustra o cálculo do valor de ativação, tal que

$$y_j^p = f_j(\text{net}_j^p) \quad (7)$$

onde:

$$\text{net}_j^p = \sum_{i=1}^n x_i^p w_{ji} \quad (8)$$

A constante n representa o número de conexões de entrada do neurônio j para camada seguinte e w_{ji} , os pesos entre estas conexões.

O algoritmo de retropropagação do erro apresenta uma série de dificuldades ou deficiências que desestimula uma maior disseminação do seu uso. O principal problema diz respeito à lentidão do algoritmo para superfícies mais complexas. Não é raro o algoritmo convergir para mínimos locais. Os mínimos locais são pontos na superfície de erro que apresentam uma solução estável, embora não sejam a saída correta. Desde que o algoritmo de retropropagação do erro foi popularizado algumas pesquisas vêm buscando técnicas tanto para acelerar o algoritmo quanto

para reduzir a incidência de mínimos locais. Dentro destas técnicas destaca-se o algoritmo de Levenberg-Marquardt (HAGAN e MENHAJ, 1994).

O algoritmo de Levenberg-Marquardt é eficiente quando se trata de redes que não possuem mais do que algumas centenas de conexões a serem ajustadas (HAGAN e MENHAJ, 1994). Ele trabalha de forma a evitar mal-condicionamento de inversão de matrizes. A seguir são descritos alguns dos algoritmos de treinamento entre eles: Levenberg-Marquardt, método Bayesiano com regularizações e método *quase*-Newton.

3.5 Levenberg-Marquardt (LM)

O treinamento LM é uma função que atualiza os pesos e valores dos bias de acordo com a otimização LM. Frequentemente é considerado como o mais rápido dos algoritmos de treinamento retropropagação do erro, porém requer mais memória computacional do que os outros algoritmos.

De acordo com Lera e Pinzolas (2002), as características principais do algoritmo LM são as seguintes:

No algoritmo LM, as mudanças (Δ) nos pesos (\bar{w}) são obtidas solucionando,

$$\alpha \Delta = -\frac{1}{2} \nabla E \quad (9)$$

onde E é o erro médio quadrático da rede.

$$E = \frac{1}{N} \sum_{k=1}^N [\bar{y}(x_k) - \bar{d}_k]^2 \quad (10)$$

onde N é o número de exemplos, $\bar{y}(x_k)$ é a saída da rede correspondente ao exemplo x_k , e \bar{d}_k é a saída desejada para aquele exemplo.

Os elementos da matriz α são dados por:

$$\alpha_{ij} = (1 + \lambda \delta_{ij}) \sum_{r=1}^p \sum_{k=1}^N \left[\frac{\partial y_r(x_k)}{\partial w_i} \frac{\partial y_r(x_k)}{\partial w_j} \right] \quad (11)$$

onde p é o número de saídas da rede. Começando com pesos iniciais aleatórios, ambos α e ∇E são calculados resolvendo a Equação 9. A correção para os valores dos pesos é obtida por ($\bar{w}' = \bar{w} + \Delta$), conhecido como época de aprendizado LM. Cada iteração com essas épocas reduz o erro até encontrar um mínimo. A variável λ na Equação 11 é o parâmetro que é ajustado a cada época de acordo com a evolução do erro. Mais detalhes de equacionamento do algoritmo LM podem ser encontrados, por exemplo, em (HAGAN e MENHAJ, 1994).

3.6 Método Bayesiano com regularizações

O método Bayesiano com regularizações é uma função que atualiza os pesos e valores dos bias de acordo com a otimização LM descritas nas Equações 9, 10 e 11. Tipicamente os algoritmos de treinamento têm como objetivo reduzir a soma dos erros quadráticos, entretanto a regularização adiciona mais um termo. Ela minimiza a combinação dos erros quadráticos e os pesos, para então produzir uma rede generalizada e com respostas mais suaves (MACKAY, 1992; FORESSE e HAGAN, 1997).

A função a ser minimizada neste caso é descrita pela seguinte equação (FORESSE e HAGAN, 1997):

$$F = (\lambda(SSE) + \delta(SSW)) \quad (12)$$

onde λ e δ são os parâmetros ajustáveis no contexto do método Bayesiano de regularização, SSE é o somatório dos erros quadráticos e SSW o somatório dos pesos quadráticos. A solução e atualização dos dois parâmetros ajustados são realizadas aplicando a lei de Bayes (ver, por exemplo, MACKAY, 1992).

Os três métodos de treinamento apresentados estão presentes na NEURAL NETWORK TOOLBOX do *software* MATLAB o qual é utilizado para os experimentos propostos neste trabalho.

4. EXPERIMENTO PRÁTICO E CARACTERÍSTICAS DAS REDES NEURAIS UTILIZADAS

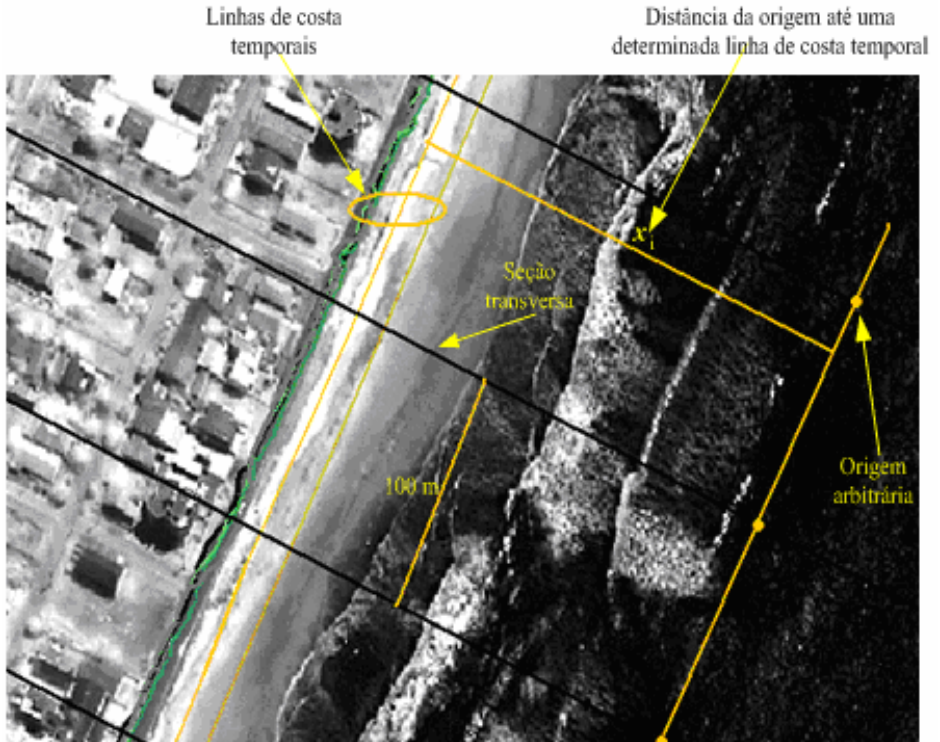
Inicialmente é necessário obter informações para servir de dados de entrada para as redes neurais. Nesta etapa todos os dados históricos foram colocados em camadas no *AutoCad*. Feito isto, foi escolhida, uma das linhas de costa para servir de referência para serem traçadas seções transversais a esta. No caso a primeira linha temporal obtida por GPS (2001) foi selecionada para ser como referência e a partir dela são traçadas seções transversais com uma distância fixa escolhida de 100 m. Para este estudo de caso foram desenhadas 57 linhas perpendiculares a linha de referência.

As coordenadas dos pontos de interseção das linhas temporais com as seções transversas são armazenadas. Com as coordenadas dos pontos de interseção são calculadas as distâncias em metros a partir de uma origem arbitrária para cada ponto. A Figura 7 apresenta os elementos auxiliares descritos anteriormente sobre uma imagem de satélite Ikonos em parte da área de estudo.

Os testes utilizando redes neurais estão divididos em dois. Todos com a característica de treinamento não-supervisionado. Ou seja, são fornecidas apenas informações para rede sobre as distâncias da linha de costa em relação à origem, formando um vetor de neurônios de entrada conforme cada seção transversal. A

rede aprende relações conforme estas informações de entrada e descobre qual a saída para passos adiante que seria a predição de 2008. Os dados de 2008 não participam do treinamento e servem como controle para verificar a resposta encontrada pelo modelo preditivo. Destaca-se que nenhum tipo de peso é selecionado para os dados de entrada. Apresentam-se a seguir as características dos testes realizados.

Figura 7 – Elementos auxiliares para coleta de dados de entrada para RNA.



4.1 Teste 1 COM RNA – MLP – Método Bayesiano com regularizações

No primeiro teste utilizou-se a arquitetura MLP e o Método Bayesiano com regularizações como algoritmo de treinamento, de acordo com as características descritas no Quadro 1.

Quadro 1 – Teste 1 com RNA, MLP e Bayesiano com regularizações.

Teste 1 - Arquitetura MLP	
Rede Neural Artificial	MLP
Método de Treinamento	Bayesiano com regularizações
Número de camadas escondidas	1
Número de neurônios escondidos	2,4,5,10
Função de ativação utilizada na camada escondida	tangente hiperbólica
Função de ativação utilizada na camada de saída	linear

4.2 Teste 2 - RNA – Elman – Quase-Newton

No segundo teste utilizou-se a arquitetura de Elman com o algoritmo de treinamento método quase-Newton, de acordo com as características descritas no Quadro 2:

Quadro 2 – Teste 2 com RNA, Elman e quase-Newton.

Teste 2 - Elman	
Rede Neural Artificial	Elman
Método de Treinamento	quase-Newton
Número de camadas escondidas	1
Número de neurônios escondidos	2,4,5,10
Função de ativação utilizada na camada escondida	Tangente hiperbólica
Função de ativação utilizada na camada de saída	linear

4.3 Critérios de avaliação

Para avaliar a eficiência dos três métodos de predição, foram calculados:

- (1) A média aritmética do desvio da linha de costa usada como controle conforme o levantamento efetuado por GPS em 2008. Isto foi calculado usando:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n (GPS2008_i - Predição2008_i) \quad (13)$$

onde $(GPS2008_i - Predição2008_i)$ denota a distância de linha de costa predita em relação à linha de controle na i -ésima seção transversal, e n o número de seções.

- (2) O desvio padrão, medida mais comum da dispersão estatística, foi obtido através da raiz quadrada da variância da posição da linha de costa estimada através da seguinte equação:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (GPS2008_i - Pr\ edição2008_i - \bar{x})^2} \quad (14)$$

- (3) O erro médio quadrático médio, RMS (*root mean square*) da linha de costa é calculado para avaliar a respectiva eficácia do método. O RMS pode ser escrito como,

$$RMS = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (GPS2008_i - Pr\ edição2008_i)^2} \quad (15)$$

- (4) O coeficiente de multicorrelação R^2 foi calculado para verificar o ajuste dos dados preditos com a linha de costa de controle usando a seguinte equação (SCHAIBLE e LEE, 1996):

$$R^2 = 1 - \frac{\sum_{i=1}^n (GPS2008_i - Pr\ edição_i)^2}{\sum_{i=1}^n (GPS2008_i - médiaGPS2008)^2} \quad (16)$$

- (5) Finalmente, para verificar o desempenho dos modelos preditivos foi quantificado o erro de percentagem média absoluta (*mean absolute percentage error*) (MAPE), que representa uma estatística da medida de acurácia no ajustamento de dados de séries temporais, especialmente em predições e usualmente é expressado em termos de percentagem (HAYATI E SHIRVANY, 2007):

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{GPS2008_t - Pr\ edição2008_t}{GPS2008_t} \right| \quad (17)$$

5. RESULTADOS E DISCUSSÕES

O ano de 2008 foi selecionado para testar as modelagens em questão, ou seja, é retirada da modelagem a linha temporal de 2008 e posteriormente é efetuada uma predição para 2008 com o objetivo de comparar os resultados entre a linha de costa predita e observada por GPS em 2008.

A seguir os resultados e análises experimentais são apresentados:

O Quadro 3 apresenta os resultados para a rede neural artificial MLP, com o algoritmo de treinamento Bayesiano com regularizações. Os resíduos absolutos entre a diferença dos valores preditos e os valores reais de GPS 2008 utilizado como controle para o conjunto de 57 amostras utilizadas no estudo são apresentados. Este quadro está dividido em cinco colunas que representam diferentes processamentos variando os neurônios na camada oculta em 2, 4, 5 e 10 para verificar o comportamento quando este parâmetro é modificado.

Quadro 3 – Resultados do teste 1 com RNA, MLP e método Bayesiano com regularização.

Teste RN 1- MLP – método Bayesiano com regularização, retropropagação do erro				
Neurônios na camada oculta				
	2	4	5	10
seções transversais	Redes Neurais (GPS2008-Predição) valor absoluto (m)	Redes Neurais (GPS2008-Predição) valor absoluto (m)	Redes Neurais (GPS2008-Predição) valor absoluto (m)	Redes Neurais (GPS2008-Predição) valor absoluto (m)
1	3,52	479,37	1,59	178,22
2	3,87	482,75	1,24	176,67
3	2,08	360,16	7,19	184,03
4	2,32	293,60	2,84	177,93
5	1,49	332,84	6,65	182,67
6	1,33	372,47	3,82	178,6
7	0,18	297,5	4,99	184,13
8	0,01	374,66	5,16	183,26
9	4,60	371,26	0,56	174,85
10	3,28	373,25	1,88	177,28
11	3,33	4,04	3,27	4,26
12	2,01	1,31	1,91	1,09
13	1,3	0,60	1,11	0,38
14	2,17	1,47	2,23	1,25
15	0,24	0,44	0,19	0,66
16	2,79	2,10	2,71	1,87
17	0,32	0,37	0,37	0,59
18	1,75	1,07	2,03	0,85
19	1,81	1,14	1,68	0,92
20	0,71	0,07	0,59	0,15
21	0,17	0,49	0,02	0,71
22	0,20	0,83	0,08	1,05
23	0,36	0,29	0,23	0,51

24	1,16	1,79	1,15	2,01
25	0,56	1,21	0,57	1,44
26	4,75	8,50	5,50	5,55
27	0,65	4,63	1,62	1,65
28	0,73	4,48	1,47	1,52
29	0,12	3,31	0,33	0,37
30	1,42	4,93	1,94	1,99
31	2,51	6,02	3,03	3,08
32	2,21	0,11	2,85	2,8
33	3,09	1,97	4,86	4,81
34	2,00	0,39	2,55	2,51
35	0,20	3,25	0,32	0,37
36	12,06	12,36	12,06	0,51
37	14,18	13,88	14,18	26,16
38	16,50	16,17	16,5	29,06
39	2,36	2,03	2,36	14,98
40	1,92	1,56	1,92	15,58
41	2,87	2,52	2,87	16,62
42	2,34	1,99	2,34	16,07
43	695,61	5,06	9,46	7,41
44	890,76	1,51	2,08	0,58
45	702,49	11,92	18,40	16,37
46	703,88	12,11	16,51	14,47
47	704,79	3,33	16,29	14,24
48	699,73	1,01	14,29	12,24
49	703,53	3,04	15,21	13,15
50	708,18	2,25	15,01	12,94
51	707,97	2,97	15,45	13,39
52	13,70	13,85	14,21	13,78
53	12,72	12,6	12,51	12,52
54	4,57	4,44	4,36	4,37
55	2,67	2,55	2,46	2,47
56	1,73	1,61	1,53	1,54
57	3,11	3,24	3,32	3,31

Os valores do Quadro 4 foram calculados utilizando as Equações 15, 16, 17, 18 e 19 seguidos dos valores mínimos e máximos para os dados encontrados no Quadro 3.

Quadro 4 – Resultados estatísticos para o teste 1.

Resultados estatísticos para o Teste 1 com RN - MLP – Método Bayesiano com regularização				
Neurônios na camada oculta	2	4	5	10
Máximo (m)	890,76	482,75	18,40	184,13
Mínimo (m)	0,01	0,07	0,02	0,15
Média Aritmética (m)	121,13	71,27	5,22	38,13
Desvio Padrão (m)	270,27	146,35	5,61	67,73
RMS (m)	288,72	158,72	7,50	75,82
R ²	0,00%	0,00%	99,38%	36,80%
MAPE	9,56%	6,75%	0,46%	3,57%

O Quadro 5 apresenta os resultados para o teste 2 com a arquitetura de Elman e o o algoritmo de treinamento quase-Newton.

Quadro 5 – Resultados do teste 2 com RNA, Elman e quase-Newton.

Teste 2 – Elman – quase-Newton				
Neurônios na camada oculta				
	2	4	5	10
seções transversais	Redes Neurais (GPS2008-Predição) valor absoluto (m)	Redes Neurais (GPS2008-Predição) valor absoluto (m)	Redes Neurais (GPS2008-Predição) valor absoluto (m)	Redes Neurais (GPS2008-Predição) valor absoluto (m)
1	3,54	6,19	3,46	3,96
2	3,89	6,92	4,19	3,30
3	2,06	0,00	2,73	5,44
4	2,33	2,84	0,08	0,51
5	1,48	1,52	1,25	2,90
6	1,35	3,76	1,00	2,33
7	0,19	1,22	1,55	1,94
8	0,03	0,86	1,90	5,07
9	4,62	1,91	0,86	3,98
10	3,30	2,91	0,14	3,12
11	3,99	0,56	0,61	1,55
12	1,36	0,57	0,53	0,49
13	0,65	0,17	0,12	0,84

14	1,52	1,34	1,38	2,02
15	0,40	1,48	1,53	2,18
16	2,14	5,91	5,87	4,40
17	0,32	0,49	0,54	1,63
18	1,12	0,86	0,91	1,29
19	1,19	0,10	0,06	0,16
20	0,12	0,32	0,36	0,36
21	0,44	1,80	1,85	2,19
22	0,78	2,40	2,45	2,71
23	0,25	0,78	0,82	1,49
24	1,74	1,61	1,66	1,88
25	1,17	1,39	1,44	2,05
26	7,86	7,20	7,20	0,16
27	3,98	4,66	4,66	0,18
28	3,84	3,43	3,43	0,23
29	2,66	2,97	2,97	0,64
30	4,28	4,16	4,16	0,06
31	5,38	5,03	5,03	1,10
32	0,54	0,34	0,34	0,73
33	2,61	4,67	4,67	10,38
34	0,25	3,56	3,56	2,18
35	2,62	16,79	16,79	0,13
36	12,09	10,14	10,14	10,14
37	14,15	0,06	0,06	0,06
38	16,46	7,13	7,12	7,12
39	2,32	1,21	1,20	1,20
40	1,88	0,39	0,39	0,39
41	2,83	1,84	1,84	1,84
42	2,3	2,57	2,56	2,56
43	7,42	2,05	7,66	2,32
44	0,56	1,31	13,89	1,05
45	16,38	6,38	3,24	4,78
46	14,48	10,85	1,18	10,41
47	14,25	13,41	3,70	9,66
48	12,25	10,95	1,24	6,75
49	13,16	12,50	2,77	5,95
50	12,95	12,41	2,64	6,23
51	13,4	8,67	1,08	2,89

52	13,57	12,37	12,37	12,37
53	12,51	11,63	11,63	11,47
54	4,35	2,10	2,10	3,54
55	2,45	1,89	1,89	2,91
56	1,52	0,49	0,49	0,05
57	3,33	1,35	1,35	0,60

O Quadro 6 apresenta os resultados estatísticos para o teste 2.

Quadro 6 – Resultados estatísticos para o teste 2.

Resultados estatísticos para o Teste 2 – Elman – quase-Newton				
Neurônios na camada oculta	2	4	5	10
Máximo (m)	16,46	16,79	16,79	12,37
Mínimo (m)	0,03	0,00	0,06	0,06
Média Aritmética (m)	4,8	4,19	3,25	3,22
Desvio Padrão (m)	5,12	4,25	3,68	3,21
RMS (m)	6,87	5,84	4,80	4,45
R ²	99,48%	99,62%	99,74%	99,78%
MAPE	0,42%	0,37%	0,29%	0,28%

Analisando-se os resultados é possível chegar às seguintes conclusões:

a) A escolha da arquitetura da rede, do modelo de treinamento e dos neurônios nas camadas ocultas são fatores importantes e influenciam na resposta do modelo de predição.

c) O teste 1, que apresenta a RN –MLP através do método de treinamento Bayesiano com regularização, teve variações significativas quando modificados os números de neurônios nas camadas ocultas e o melhor resultado encontrado foi para 5 neurônios com as seguintes estatísticas quando comparada com a linha de costa de controle (máximo valor; mínimo valor; média aritmética; desvio padrão; RMS; R²; MAPE), (18,40 m; 0,02 m; 5,22 m; 5,61 m; 7,50 m; 99,38%; 0,46%). Os demais testes para este caso fugiram completamente da realidade.

d) Para o teste 2 com a rede de Elman, obteve-se pequenas variações quando são modificados os neurônios na camada oculta com o MAPE variando de 0,28% para 10 neurônios na camada oculta; 0,29% para 5; 0,37% para 4 e 0,42% para 2 conforme o QUADRO 6.

e) Os melhores resultados encontrados com RN de Elman, foram com 10 neurônios na camada oculta apresentando os seguintes resultados (máximo valor; mínimo valor; média aritmética; desvio padrão; RMS; R²; MAPE), (12,37 m; 0,06 m; 3,22 m; 3,21 m; 4,45 m; 99,78%; 0,28%).

Douglas e Crowell (2000) apontam que é viável a elaboração de modelos preditivos com precisão no nível de dezenas de metros, pois se deve considerar a variabilidade inerente da posição da linha de costa. No trabalho de Crowell *et al.* (1991) estima-se que o erro posicional envolvido na extração da linha de costa a partir de fotografias aéreas verticais históricas é em torno de 8 m.

6. CONCLUSÕES

O primeiro teste com as redes neurais teve a intenção de utilizar a arquitetura MLP e o treinamento Bayesiano com regularizações. Neste teste com 2, 4 e 10 neurônios nas camadas ocultas os resultados foram muito distantes da linha de controle. Porém com 5 neurônios na camada oculta encontrou-se um resultado considerado bom apresentando um uma média aritmética de 5,22 m com desvio-padrão de 5,61 m e o resíduo máximo de 18,40 m.

No segundo teste utilizando a rede de Elman e o treinamento quase-Newton encontrou-se um resíduo máximo de 12,37 m e uma média aritmética de 3,22 m com desvio-padrão de 3,21 m, para o caso de 10 neurônios na camada oculta.

Os parâmetros R^2 e MAPE indicam que a melhor correlação entre a linha de costa predita de 2008 com a linha de costa de controle para o mesmo ano ocorreu com o modelo utilizando o teste 2 com a rede de Elman, obtendo os seguintes valores: 99,78% para o R^2 e 0,28% para o MAPE.

Destacam-se dois problemas referentes à qualidade dos dados temporais que normalmente ocorrem com modelos preditivos de linha de costa:

1. Os dados são pobremente amostrados.
2. Não existe uma regularidade entre os espaçamentos na aquisição dos dados temporais.

Mesmo levando-se em conta esses problemas citados anteriormente e as incertezas no processo de extração de linha de costa através de documentos cartográficos antigos, as redes neurais artificiais são uma boa alternativa quando várias configurações podem ser testadas em busca de um modelo preditivo para linha de costa.

AGRADECIMENTOS

RMG agradece a bolsa de doutorado do CNPq e a bolsa CAPES recebida durante o período de doutorado sanduíche pelo projeto PROBRAL realizado no GIK (Geodetic Institute Karlsruhe), Alemanha.

REFERENCIAS BIBLIOGRÁFICAS

- ARBIB, M. A. *The handbook of brain theory and neural networks*. Massachusetts Institute of Technology, 2003. 1309p.
- BRAGA, A. DE P.; CARVALHO A. P. DE L. F.; LUDEMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. LTC - Livros Técnicos e Científicos Editora S.A., 2000. 262p.

- CROWELL, M.; LETHERMAN, S. P.; BUCKLEY, M. K.; Historical shoreline change: error analysis and mapping accuracy. *Journal of Coastal Research*, v. 7, n. 3, 1991. p. 839-852.
- DIAMAT. Determinação do Impacto Ambiental Decorrente da Ação Antrópica na Linha Costeira da Região de Matinhos. *Relatório Técnico*. Coordenador: Prof. Dra. Cláudia Pereira Krueger. Universidade Federal do Paraná, 2000.
- DOUGLAS, B. C.; CROWELL, M. Long-term shoreline position prediction and error propagation. *Journal of Coastal Research*, v. 16, n. 1, 2000. p. 145-152.
- FLETCHER, R.; POWELL, M. J. D. A rapidly convergent descent method for minimization, *Computer Journal*, v. 7, 1963. p. 149-154.
- FORESEE, F. D.; HAGAN, M. T. Gauss-Newton Approximation to Bayesian Regularization, *International Joint Conference on Neural Networks*, 1997. p. 1930-1935.
- FUJII, S. Neural network signal processing for HF ocean radar. *International Geoscience and Remote Sensing Symposium of Earth Environment, IGARSS '93 Better Understanding of Earth Environment*, v. 1, 1993. p. 212-214.
- GEROMEL, L. H. *Aplicação de sistemas inteligentes em projetos de transformadores de potência*. Tese (Doutorado), Faculdade de Engenharia Elétrica e de Computação. UNICAMP, 2003. 144p.
- GUERRA F. A. *Análise de métodos de agrupamento para o treinamento de redes neurais de base radial aplicadas à identificação de sistemas*. Dissertação (Mestrado) Programa de Pós-Graduação em Engenharia de Produção e Sistemas. PUC-PR, 2006. 131p.
- GOLDFARB, D. A family of variable metric methods derived by variational means, *Mathematics of Computation*, v. 24, 1970. p. 23-26.
- GONÇALVES, R. M.; KRUEGER, C. P.; TANAJURA, E. L. X.; LEANDRO, D.; HUINCA, S. M. Mapeamento Costeiro Cadastral de Linha de Costa. In: COBRAC, 2008, Florianópolis. *Anais do VIII Congresso de Cadastro Técnico Multifinalitário e Gestão Territorial*, 2008.
- HAGAN, M. T.; MENHAJ M. B. Training feedforward networks with the Marquardt algorithm. *IEE Transactions on Neural Networks*, v. 5, n. 6, 1994. p. 989-993.
- HAYATI, M.; SHIRVANY Y. Artificial neural network approach for short term load forecasting for Illam region. *World Academy of Science, Engineering and Technology*, n. 28, 2007. p. 280-284.
- HAYKIN, S. *Neural Networks – A comprehensive foundation*. McMaster University Hamilton, Ontario, Canada, Pearson Education, 1999. 823p.
- KRÖSE, B.; SMAGT P. V. D. *An introduction to Neural Networks*. The University of Amsterdam. Eighth edition, 1996. 136p.

- KRUEGER, C. P.; GONÇALVES, R. M.; HECK, B. Surveys at the Coast of Parana, Brazil, to Determinate the Temporal Coastal Changes. *Journal of Coastal Research*, v. 1, 2009. p.632-635.
- KUBOTA, N.; HASHIMOTO, S.; KOJIMA, F. A sensory network for perception-based robotics using neural networks. *IEEE International Joint Conference on Neural Networks*, v. 4, 2003. p. 3151-3156.
- LERA, G.; PINZOLAS M. Neighborhood based Levenberg-Marquardt algorithm for neural network training. *IEE Transactions on Neural Networks*, v. 13, n. 5, 2002. p. 1200-1203.
- MACIEL, J. *Análise de um sistema de crédito cooperativo através de redes neurais (MLP) com a utilização do algoritmo Levenberg Marquardt*. Dissertação (Mestrado) Programa de Pós-Graduação em Métodos Numéricos em Engenharia – UFPR, 2005. 87p.
- MACKAY, D. J. C. Bayesian interpolation, *Neural Computation*, v. 4, 1992. p. 415-447.
- MITISHITA, E. A. *Monorrestituição digital de fotos associada com sistema de computação gráfica CAD, para fins de mapeamento na área florestal*. Tese (Doutorado) Programa de Pós-Graduação em Engenharia Florestal, UFPR, 1997.
- NEURAL NETWORK TOOLBOX, *The MathWorks™ - Accelerating the pace of engineering and science*. Comunicação do software Matlab disponível em: <http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/index.html?access/helpdesk/help/toolbox/nnet/function.html#9361> Acesso em: 05/06/2009
- SCHAIBLE, B.; LEE, Y. C. Fuzzy logic models with improved accuracy and continuous differentiability. *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, Part C, v. 19, n. 1, 1996. p. 37-47.
- SHANNO, D. F. Conditioning of quasi-Newton methods for function minimization, *Mathematics of Computation*, v. 24, 1970. p. 145-160.
- SOUZA, C. R. G. Coastal erosion and the coastal zone management challenges in Brazil. *Journal of Integrated Coastal Zone Management*. v. 9, n.1, 2009. p. 17-37.
- SOUZA, E. C. B. *Metodologia para modelagem do relevo oceânico usando redes neurais artificiais*. Tese (Doutorado) Programa de Pós-Graduação em Ciências Geodésicas – UFPR, 2006. 208p.
- TAN, G.; HUIMIN S.; WANG F.; DENG C. Short-term traffic flow prediction based on parallel quasi-Newton neural network, *International Conference on Measuring Technology and Mechatronics Automation*, v. 3, 2009. p. 305-308.
- VAFEIDIS, A. T.; KOUKOULAS, S.; GATSIS, I.; GKOLTSIU, K. Forecasting land-use changes with the use of neural networks and GIS, *Geoscience and Remote Sensing Symposium. IEEE International*, 2007. p. 5068-5071.

- WALDEMARK, J.; DOVNER P. O.; KARLSSON J. Hybrid neural network pattern recognition system for satellite measurements. *IEEE International Conference on Neural Networks*, v. 1, n. 1, 1995. p. 195-199.
- WANDRESEN, R. *Integração entre redes neurais e correlação para identificar pontos de apoio pré-sinalizados*. Tese (Doutorado) Programa de Pós-Graduação em Ciências Geodésicas – UFPR, 2004. 141p.
- WINDROW, B.; HOFF M. E. Adaptive switching circuits. Institute of Radio Engineers, *Western Electronic Show and Convention*, 1960 p. 96-104.

Recebido em dezembro de 2010. Aceito em abril de 2010)