

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Ajustes do Modelo Processo Estocástico  
Neural aplicado às séries de Energia Natural  
Afluente**

**Diego Ramalho de Oliveira**

JUIZ DE FORA  
JULHO, 2014

# Ajustes do Modelo Processo Estocástico Neural aplicado às séries de Energia Natural Afluente

DIEGO RAMALHO DE OLIVEIRA

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Luciana Conceição Dias Campos

JUIZ DE FORA

JULHO, 2014

# AJUSTES DO MODELO PROCESSO ESTOCÁSTICO NEURAL APLICADO ÀS SÉRIES DE ENERGIA NATURAL AFLUENTE

Diego Ramalho de Oliveira

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

---

Luciana Conceição Dias Campos  
D. Sc. em Engenharia Elétrica, PUC-Rio, 2010

---

André Luís Marques Marcato  
D. Sc. em Engenharia Elétrica, PUC-Rio, 2002

---

Heder Soares Bernardino  
D. Sc. em Modelagem Computacional, LNCC, 2012

JUIZ DE FORA  
18 DE JULHO, 2014

*Dedico aos meus pais, pelo apoio e sustento, a  
minha esposa e também a meu irmão e ami-  
gos.*

## Resumo

O modelo de processo estocástico genérico baseado em redes neurais denominado Processo Estocástico Neural (PEN) foi proposto para ser aplicado em problemas com características estocásticas e/ou periódicas. O PEN conseguiu gerar séries sintéticas com boa aderência às séries históricas de Energia Natural Afluente (ENA) no contexto do planejamento da operação do sistema hidroelétrico brasileiro. O objetivo desse trabalho é modificar o treinamento, as funções de ativação e a janela temporal das redes neurais artificiais que compõem o PEN, para gerar cenários aderentes às séries históricas de afluições mensais, as quais possuem cinco anos a mais do que as utilizadas no modelo PEN original. Os resultados revelam cenários estatisticamente tão aderentes quanto os obtidos no modelo PEN original.

**Palavras-chave:** Planejamento da operação energética, Séries de Afluições com Períodos Mensais, Redes Neurais, Processos Estocásticos, Processo Estocástico Neural.

# Abstract

The general stochastic model based on neural networks called stochastic neural process (PEN) has been proposed to be applied to problems with stochastic and / or periodic features. The PEN was able to generate synthetic series with good adherence to the historical series of Affluent Natural Energy (ENA) in the context of planning for the operation of the Brazilian hydroelectric system. This work to modify the training, the activation functions and the temporal window of neural networks comprising the PEN to generate adherent to the historical series of monthly inflows scenarios, which are five years longer than those used in the model original PEN. The results show statistically scenarios as adherent as those obtained in the original PEN model.

**Keywords:** Operation Planning Energy, Inflows Series with Monthly Periods, Neural Network, Stochastic Process, Neural Stochastic Process.

## **Agradecimentos**

A todos os meus parentes, pelo encorajamento e apoio.

A professora Luciana pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

*“Escolhe um trabalho de que gastes, e não terás que trabalhar nem um dia na tua vida”.*

*Confúcio*



# Sumário

<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Abreviações</b>	<b>8</b>
<b>1 Introdução</b>	<b>9</b>
1.1 Apresentação do tema e contextualização do problema . . . . .	9
1.2 Justificativa . . . . .	11
1.3 Objetivos . . . . .	12
1.3.1 Objetivo geral . . . . .	12
1.3.2 Objetivo específico . . . . .	12
<b>2 Fundamentos Teóricos</b>	<b>13</b>
2.1 Processo estocástico . . . . .	13
2.2 Redes neurais artificiais . . . . .	14
2.3 O modelo PEN . . . . .	19
<b>3 Metodologia</b>	<b>23</b>
<b>4 Resultados</b>	<b>28</b>
4.1 Análise do ajuste do novo modelo PEN . . . . .	28
4.2 Análise dos cenários sintéticos gerados . . . . .	29
<b>5 Considerações finais</b>	<b>34</b>
<b>Referências Bibliográficas</b>	<b>35</b>
<b>A Resultados adicionais</b>	<b>37</b>
A.1 Melhores CE de cada subsistema . . . . .	37

## Lista de Figuras

1.1	Regiões das empresas do SIN: Subsistemas S (Sul), Subsistemas SE/CO (Sudeste e Centro-Oeste), Subsistemas N (Norte) e Subsistemas NE (Nordeste) (ANNEL, 2014). . . . .	10
1.2	Séries Históricas de cada subsistema do SIN. . . . .	11
2.1	Energia Natural Afluente X Meses. . . . .	13
2.2	Modelo simplificado de um neurônio biológico (Wasserman, 1989). . . . .	15
2.3	Modelo simplificado de um neurônio artificial (Haykin, 2001). . . . .	15
2.4	Funções de ativação mais utilizadas: (a) função degrau; (b) função linear; (c) função logística; (d) tangente hiperbólica; (Fiorin et. al, 2011). . . . .	16
2.5	Exemplo de uma rede neural <i>feedforward</i> (Haykin, 2001). . . . .	17
2.6	Exemplo de uma rede neural recorrente (Haykin, 2001). . . . .	18
2.7	PEN ajustado para os subsistemas do SIN (Campos, 2010). . . . .	20
2.8	Componente Estocástica do PEN (Campos, 2010). . . . .	20
2.9	Encadeamento das CEs do PEN (Campos, 2010). . . . .	21
3.1	Pseudo-código do treinamento e da geração dos cenários . . . . .	27

## Lista de Abreviações

ANEEL	Agência Nacional de Energia Elétrica
CE	Componentes Estocásticas
DCC	Departamento de Ciência da Computação
ENA	Energia Natural Afluente
MAPE	Erro percentual médio absoluto
MLP	<i>Multilayer Perceptron</i>
ONS	Operador Nacional do Sistema Elétrico
PEN	Processo Estocástico Neural
RNA	Redes Neurais Artificiais
SIN	Sistema Interligado Nacional
UFJF	Universidade Federal de Juiz de Fora

# 1 Introdução

## 1.1 Apresentação do tema e contextualização do problema

O planejamento da operação do sistema hidrotérmico brasileiro é de extrema importância, já que atualmente, segundo a Agência Nacional de Energia Elétrica(ANEEL), o parque gerador brasileiro é predominantemente hidráulico (ANNEE, 2014). Além disso, esse planejamento é um problema de grande porte e de alto nível de complexidade, necessitando da divisão em diversas etapas com diferentes períodos e horizontes distintos (médio prazo, curto prazo e programação diária). Na etapa de médio prazo, foco de estudo deste trabalho, são realizados estudos visando até 5 anos à frente, tendo o objetivo de definir as parcelas de geração hidráulica e térmica que minimizam o valor esperado do custo de operação do sistema de geração e transmissão elétrica do Brasil.

No Brasil, o Operador Nacional do Sistema Elétrico (ONS) é o órgão responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no Sistema Interligado Nacional (SIN), sob a fiscalização e regulação da ANEEL. O SIN é formado por quatro subsistemas regionais, Sul (S), Sudeste/Centro-Oeste (SE/CO), Nordeste (NE) e Norte (N). A Figura 1.1<sup>1</sup> mostra os subsistemas do SIN.

Para cada subsistema do SIN existe um histórico consolidado de Energia Natural Afluente (ENA) que é disponibilizado pelo ONS desde janeiro de 1931 até dezembro de dois anos anteriores ao corrente. Por exemplo, para o ano de 2014, o ONS disponibiliza o histórico de vazões naturais para cada usina de janeiro de 1931 até dezembro de 2012. Esse trabalho foi focado até o ano de 2010. A ENA é calculada com base nas vazões naturais e nas produtividades equivalentes dos reservatórios. As séries de ENA correspondem à estimativa da energia que pode ser gerada com todas as vazões afluentes a cada um dos

---

<sup>1</sup>Os Estados do Acre (AC) e de Rondônia (RO) foram incorporados ao sistema SE/CO em 2009 (Interligação Acre-Rondônia, 2010)

reservatórios que compõem aquele reservatório equivalente. Elas possuem características não-lineares e são séries sazonais com períodos de 12 meses, levando assim a um problema estocástico não-linear (Ballini et al, 2003). A Figura 1.2 apresenta as curvas de cada ano das séries históricas de ENA de cada subsistema do SIN. Nela é possível observar a periodicidade e sazonalidade destas séries.

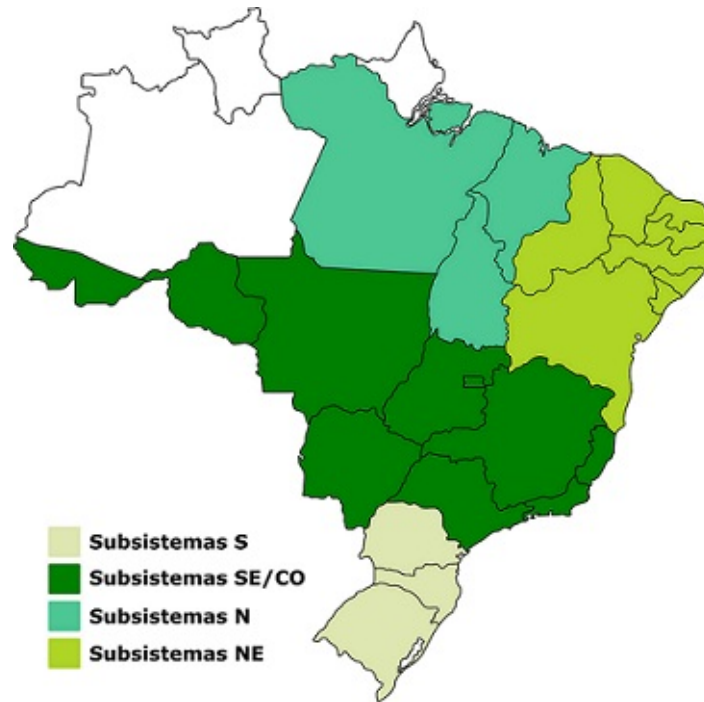


Figura 1.1: Regiões das empresas do SIN: Subsistemas S (Sul), Subsistemas SE/CO (Sudeste e Centro-Oeste), Subsistemas N (Norte) e Subsistemas NE (Nordeste) (ANNEL, 2014).

Um das formas de capturar as características das vazões dos reservatórios é a utilização de algum modelo computacional, como por exemplo: Redes Neurais Artificiais (RNA) (Haykin, 2001) e Modelos Autoregressivos (Ballini et al, 1997). Os modelos autoregressivos são modelos estatísticos lineares que tem como premissa básica que a série temporal é gerada por um processo estocástico cuja a natureza pode ser representada através de um modelo. O modelo  $AR$  (autoregressivo), o modelo  $ARMA$  (combinação dos modelos  $AR$  e de médias móveis- $MA$ ) e o  $PAR(p)$  (que é um modelo autoregressivo periódico de ordem  $p$ ) são exemplos de modelos autoregressivo. Esses modelos são limitados pois assumem que existe uma relação linear entre os elementos da série e se baseiam na hipótese de que a série em que operam é estacionária (ou seja, que a média e o desvio padrão das observações não variam com o tempo). Por outro lado, existem as RNAs que

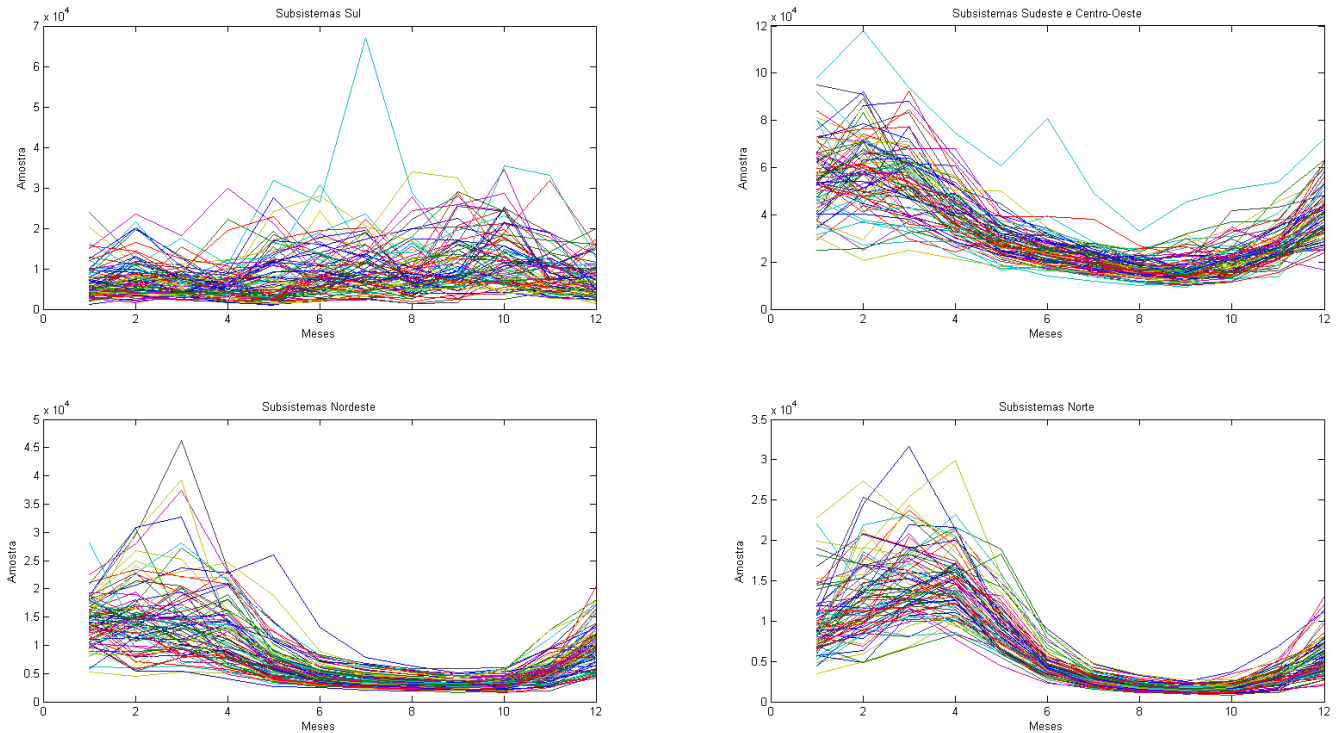


Figura 1.2: Séries Históricas de cada subsistema do SIN.

são modelos não-lineares (Haykin, 2001), de fácil entendimento e implementação que vêm sendo aplicadas em diversos problemas de análise de séries temporais ((Ballini et al, 1997), (Ballini et al, 2003), (Batista, 2009), (Campos, 2010), (Coelho et al, 2000), (Fiorin et. al, 2011)). O uso de RNAs é interessante devido à habilidade de aprendizado, capacidade de associação e generalização que elas possuem.

Neste contexto, Campos (2010) criou um modelo de processo estocástico periódico não-linear auto-regressivo genérico baseado em redes neurais, denominado Processo Estocástico Neural (PEN). Esse modelo foi aplicado no tratamento das incertezas das séries de afluências mensais através da geração de cenários hidrológicos. O PEN foi utilizado para simular cenários de ENA, sendo ajustado para cada subsistema do SIN.

## 1.2 Justificativa

A geração de cenários no contexto do planejamento da operação do sistema hidrotérmico brasileiro é muito importante, uma vez que só há disponível a série histórica, que são

os registros de afluições observados, sendo esses dados insuficientes para compor uma amostra de tamanho necessário para estimar riscos aceitáveis. A partir dos cenários é possível avaliar melhor os riscos e incertezas do sistema hidroelétrico, uma vez que eles reproduzem as características da série histórica.

Campos (2010) mostrou que os cenários gerados pelo PEN possuem boa aderência à série histórica podendo assim ser utilizado para tratar as incertezas das séries de ENA. Este trabalho consiste em aplicar algumas alterações no modelo PEN com o intuito de que ele consiga gerar cenários mais aderentes.

## **1.3 Objetivos**

### **1.3.1 Objetivo geral**

Fazer modificações no modelo PEN criado por Campos (2010), para melhor ajustá-lo às séries de ENA com o objetivo de melhorar a aderência dos cenários sintéticos gerados com as séries históricas.

### **1.3.2 Objetivo específico**

Ajustar o lag de auto-regressão, que é denominado ordem, das entradas do modelo PEN.

Aplicar uma nova série histórica de ENA que contém 5 anos a mais de dados do que os utilizados por Campos (2010), e criar assim um conjunto de teste para verificar a capacidade de generalização na etapa de ajuste do treinamento do PEN.

Aplicar diferentes algoritmos de aprendizado, visando investigar uma melhoria no treinamento das redes neurais que compõem o modelo PEN.

## 2 Fundamentos Teóricos

### 2.1 Processo estocástico

Medindo-se, por exemplo, a energia natural afluyente no subsistema SE/CO por um período de cinco anos consegue-se a curva 1 e fazendo as mesmas medições para outra parcela de cinco anos obtém-se a curva 2, ambas representadas pela figura 2.1.

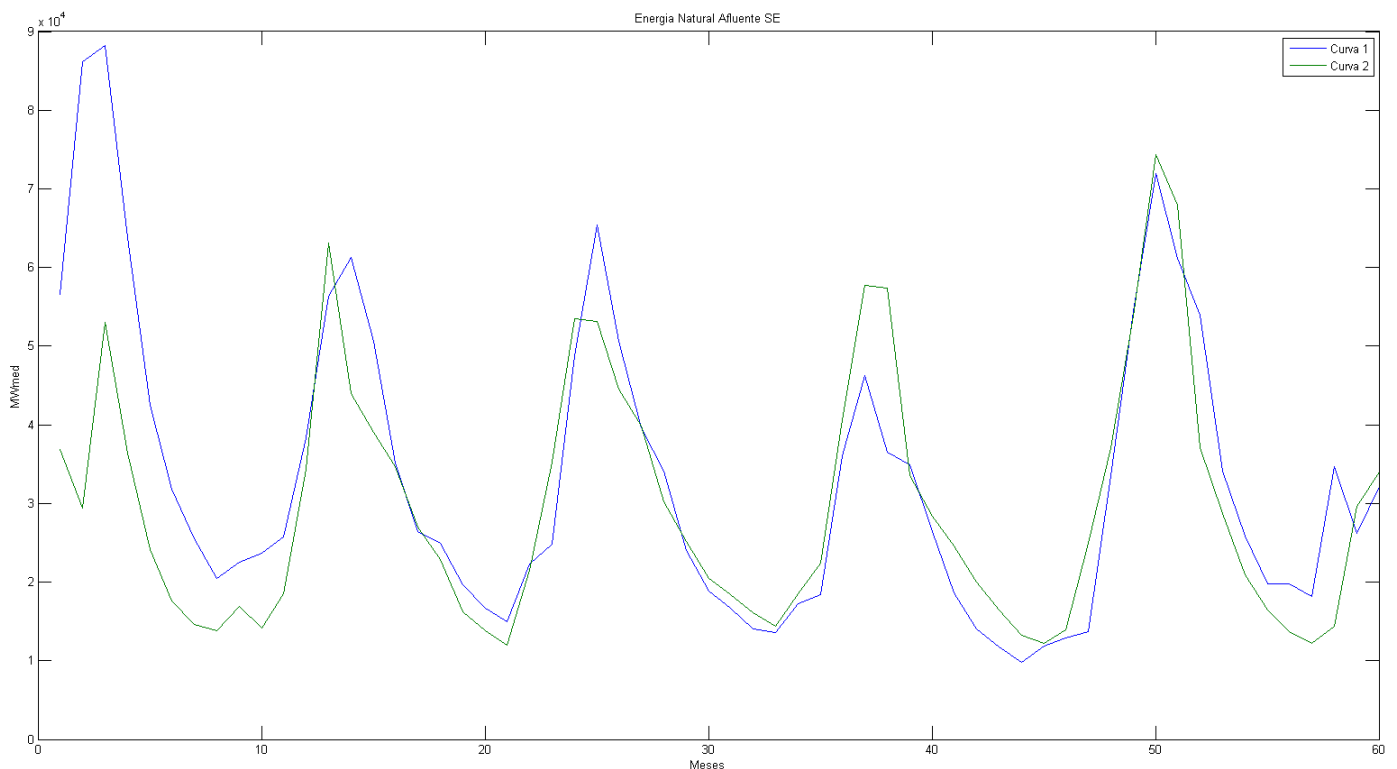


Figura 2.1: Energia Natural Afluyente X Meses.

Essas curvas são chamadas de realizações ou trajetórias do processo físico que está sendo observado e este pode ser modelado por um processo estocástico, que é simplesmente o conjunto de todas as possíveis trajetórias que podem ser observadas. Cada trajetória também é chamada de função amostral ou série temporal (Morettin et al, 2006).

Na prática, existe somente uma série temporal do processo estocástico chamada série histórica. Assim, o processo estocástico é uma variável aleatória para cada instante do tempo. O valor da série histórica no instante  $t$  é o valor “amostrado” da distribuição de probabilidade associado à variável aleatória do processo estocástico no instante  $t$  (Morettin



et al, 2006).

Em outras palavras, processo estocástico é definido como uma família de variáveis aleatórias definidas num mesmo espaço de probabilidades que descreve a evolução, no tempo ou no espaço, de algum processo físico (Papoulis, 2001). Esse conceito proporciona a análise probabilística de séries temporais. Uma série temporal consiste de um conjunto de observações realizadas sequencialmente ao longo do tempo em que existe dependência entre observações consecutivas (Souza, 2004).

Observando o comportamento da série em relação à origem dos tempos, é possível classificar os processos estocásticos de 3 tipos: aqueles que exibem propriedades de estacionariedade em longos períodos (estritamente estacionários); aqueles que possuem uma razoável estacionariedade em períodos curtos (fracamente estacionários ou estacionários de segunda ordem) e aqueles que são obviamente não estacionários. Por ser a suposição mais frequente, o conceito de estacionariedade é importante para o estudo de séries temporais. Infelizmente, uma grande parte das séries encontradas na prática são não estacionárias, podendo ser estacionarizadas através de técnicas utilizadas para remover ou filtrar a parte não estacionária, trabalhando apenas com a parte estacionária (Müller, 2007).

## 2.2 Redes neurais artificiais

Redes neurais artificiais (RNAs) são modelos computacionais não lineares, inspirados na estrutura massivamente paralela do cérebro humano, capazes de realizar as operações de aprendizado, associação, generalização e abstração através de conhecimento experimental (Haykin, 2001).

As RNAs são estruturas compostas de diversas unidades processadoras chamadas de neurônios artificiais, que são interligados através de conexões, denominadas pesos sinápticos, que armazenam o conhecimento adquirido pela rede. O neurônio artificial é inspirado no neurônio biológico. As interligações entre neurônios biológicos são efetuadas através de sinapses. As sinapses são pontos de contato controlados por impulsos elétricos e por reações químicas (devido às substâncias chamadas neurotransmissoras) entre dendritos e axônios, formando uma rede de transmissão de informações (Kovács, 2002). A Figura 2.2 mostra a estrutura de um neurônio biológico, em que as setas indicam o sentido

dos impulsos nervosos. Note que a transmissão dos impulsos nervosos de um neurônio se inicia pelos dendritos, passa pelo núcleo e termina nos axônios.

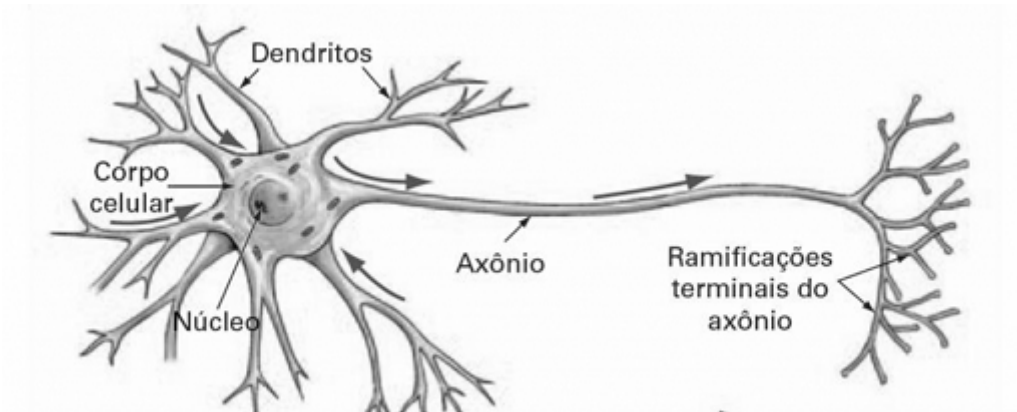


Figura 2.2: Modelo simplificado de um neurônio biológico (Wasserman, 1989).

O neurônio artificial (vide a Figura 2.3) apresenta um conjunto de entradas  $x_1, x_2, \dots, x_n$  simulando os dendritos, e uma saída  $y_j$  simulando o axônio. As entradas do neurônio são ponderadas por pesos sinápticos  $w_{j1}, w_{j2}, \dots, w_{jn}$  e somadas, fornecendo o potencial interno do processador  $V_j$ . O bias  $b_j$  é um termo de polarização do neurônio artificial, que pode ser tratado como um peso sináptico cuja entrada é sempre 1, e o seu objetivo é aumentar (ou diminuir) a influência do valor da combinação linear das entradas. A saída do neurônio  $y_j$  é obtida através da aplicação de uma função de ativação  $\varphi$ . A função de ativação (ou função restritiva)  $\varphi$  é utilizada para limitar a amplitude da saída de um neurônio (Haykin, 2001).

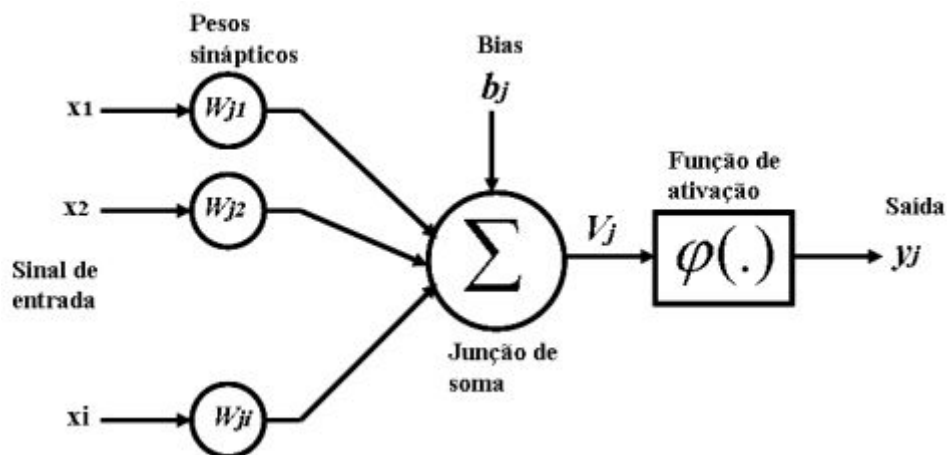


Figura 2.3: Modelo simplificado de um neurônio artificial (Haykin, 2001).

A função de ativação pode apresentar características lineares ou não lineares. As funções com comportamento linear que se destacam na literatura são: a função degrau ou limiar (Figura 2.4 (a)), onde a saída da rede será 1 se o valor  $V_j$  é  $\geq 0$  caso contrário e a função linear (Figura 2.4 (b)), que não altera o valor  $V_j$ . É comum o uso de funções de ativação do tipo sigmoide, que são funções não lineares cujo o gráfico tem a forma de S. Normalmente para mapear valores de  $V_j$  entre 0 e 1 utiliza-se a função logística (Figura 2.4 (c)) e caso se deseje estender para o intervalo de -1 a 1 se utiliza a função tangente hiperbólica (Figura 2.4 (d)).

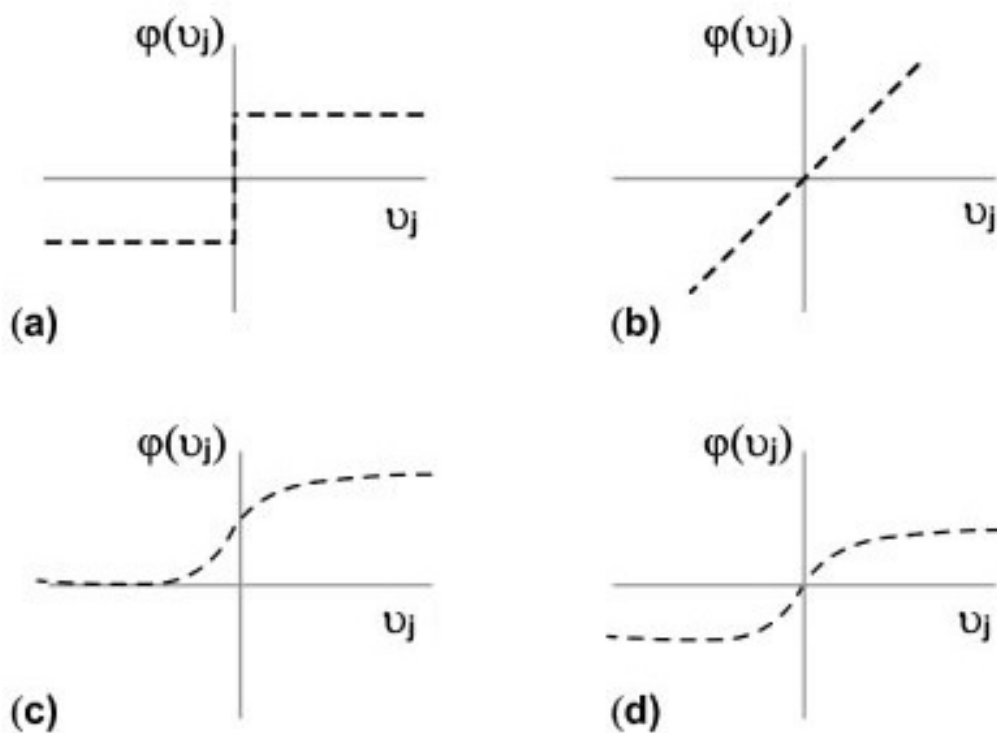


Figura 2.4: Funções de ativação mais utilizadas: (a) função degrau; (b) função linear; (c) função logística; (d) tangente hiperbólica; (Fiorin et. al, 2011).

Existem dois tipos de topologia de RNAs que são muito empregados na literatura: as Redes Neurais Não Recorrentes ou *Feedforward* e Redes Neurais Recorrentes. As Redes Neurais *Feedforward* são aquelas que não possuem conexões ligando neurônios da mesma camada ou ligando um neurônio de uma camada não consecutiva e não apresentam realimentação de suas saídas para as suas entradas (Hagan et al, 1994). Um exemplo muito usado são as redes neurais *Multilayer Perceptron* (MLP), que são redes *feedforward*

fundamentadas em *perceptron* com várias camadas. *Perceptron* é um modelo de rede neural com somente um nível que é capaz de classificar padrões que sejam linearmente separáveis. As redes MLP possuem uma ou mais camadas ocultas, além das camadas de entrada e de saída. Essas redes são normalmente usadas em estudos de séries temporais ((Batista, 2009), (Sousa et al, 2010)). A Figura 2.5 mostra uma rede neural *feedforward*. A rede possui um conjunto de entradas cujos pesos sinápticos ligam apenas os neurônios da primeira camada da RNA. A entrada dos neurônios das camadas seguintes é a saídas dos neurônios da camada imediatamente anterior.

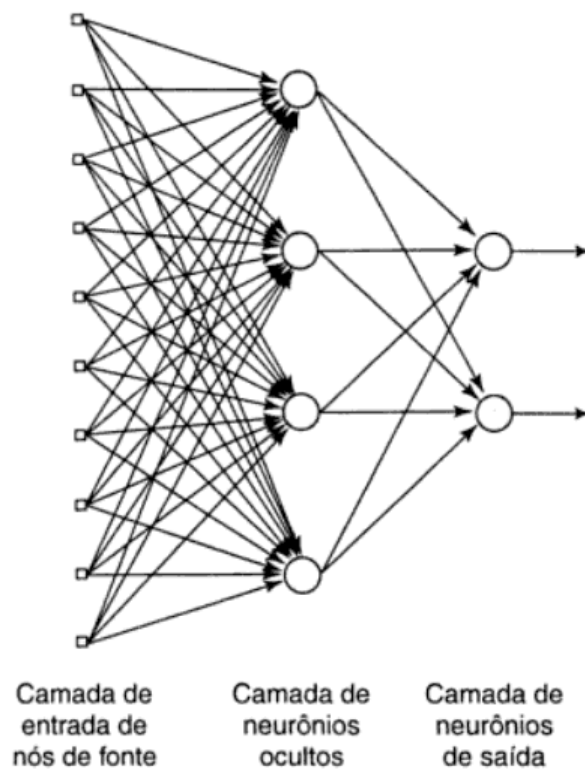


Figura 2.5: Exemplo de uma rede neural *feedforward* (Haykin, 2001).

As Redes Neurais Recorrentes são redes que contêm pelo menos um laço de realimentação das saídas para as entradas. Desta forma, as saídas destas redes são determinadas pelas entradas atuais mais as saídas anteriores. As estruturas das redes neurais recorrentes podem apresentar interligações entre neurônios da mesma camada e/ou entre neurônios de camadas não consecutivas, como ilustra a Figura 2.6 (Elman, 1990). Observe que as entradas marcadas com os operadores de atraso unitários  $z^{-1}$  são as saídas

das camadas previamente calculadas.

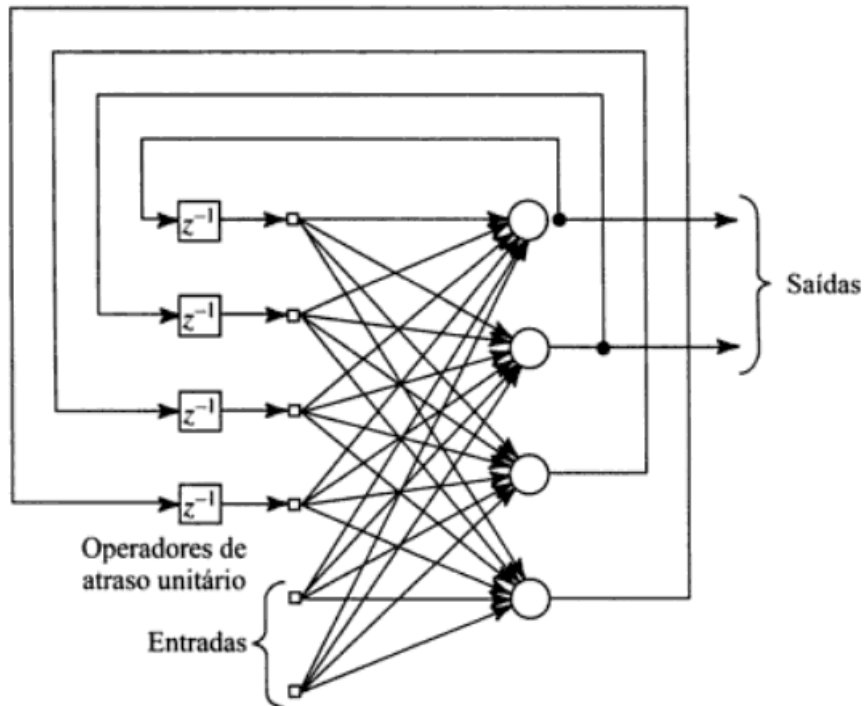


Figura 2.6: Exemplo de uma rede neural recorrente (Haykin, 2001).

Os dados coletados são normalmente separados em duas categorias: dados de treinamento, que serão utilizados para o treinamento da rede e dados de teste, que serão utilizados para verificar o seu desempenho sob condições reais de utilização. Além dessa divisão, pode-se usar também uma subdivisão do conjunto de treinamento, criando um conjunto de validação, utilizado para verificar a eficiência da rede quanto a sua capacidade de generalização durante o treinamento, e podendo ser empregado como critério de parada do treinamento.

O treinamento das redes neurais pode ser classificado de dois tipos: Treinamento Supervisionado e Treinamento não Supervisionado. No treinamento supervisionado, os padrões de dados são compostos pelos dados de entrada e as suas saídas desejadas. Ela ajusta os pesos sinápticos de modo a minimizar a diferença entre a resposta desejada e a resposta encontrada por ela, construindo assim, um mapeamento de entrada e saída. No treinamento não supervisionado, os padrões de dados não contém as saídas desejadas, somente os dados de entrada. Nesse treinamento, padrões com propriedades estatísticas semelhantes extraídas dos dados de entrada são agrupados (Haykin, 2001).

Uma das formas que são utilizadas como critério de parada do treinamento da RNA é uma função de custo da rede neural, que é calculada com os erros gerados nos neurônios da camada de saída para todos os padrões de treinamento. Uma das funções de custo mais utilizadas é o Erro Médio Quadrado (MSE), mostrada em 2.1.

$$MSE = \frac{1}{N} \sum_{j=1}^N (alvo_j - prev_j)^2 \quad (2.1)$$

onde  $N$  é a quantidade de padrões de dados,  $alvo_j$  é o valor real desejado na previsão e  $prev_j$  o valor previsto pela RNA.

## 2.3 O modelo PEN

Campos (2010) criou um modelo de processo estocástico periódico não-linear auto-regressivo genérico baseado em redes neurais, denominado Processo Estocástico Neural (PEN). Ele foi proposto para ser aplicado em problemas com características periódicas e/ou fenômenos de comportamento estocástico. Esse modelo captura as características da série histórica e gera séries temporais sintéticas igualmente prováveis à série histórica, sem necessitar de fazer suposições a priori sobre o comportamento da série ou fazer alguma decomposição da mesma, para remover ciclos ou tendências ou mesmo deixá-la estacionária.

O PEN utiliza redes neurais artificiais MLP com somente uma camada escondida, treinadas com o algoritmo de treinamento supervisionado *Levenberg-Marquardt backpropagation*<sup>2</sup>.

Ele foi aplicado no tratamento das séries de ENA e ajustado para cada subsistema do SIN. Como as séries de ENA são séries sazonais com períodos de 12 meses, os sistemas PEN foram compostos de 12 componentes estocásticas (CEs). A Figura 2.7 mostra as configurações do modelo PEN como descrito anteriormente.

Cada CE do PEN é formada por uma RNA (que é alimentada por uma janela temporal que contém os dados da série para gerar uma saída) e uma distribuição de probabilidade como ilustra a Figura 2.8. A distribuição de probabilidade é obtida através da diferença entre a saída desejada e saída real na fase do treinamento da rede neural.

---

<sup>2</sup>O algoritmo de *Levenberg-Marquardt backpropagation* é uma variação do algoritmo *backpropagation* (Hagan et al, 1994)



Figura 2.7: PEN ajustado para os subsistemas do SIN (Campos, 2010).

As saídas das RNAs são somadas com o ruído aleatório, proveniente da distribuição de probabilidade relativa à aquela CE, para gerar as séries sintéticas.

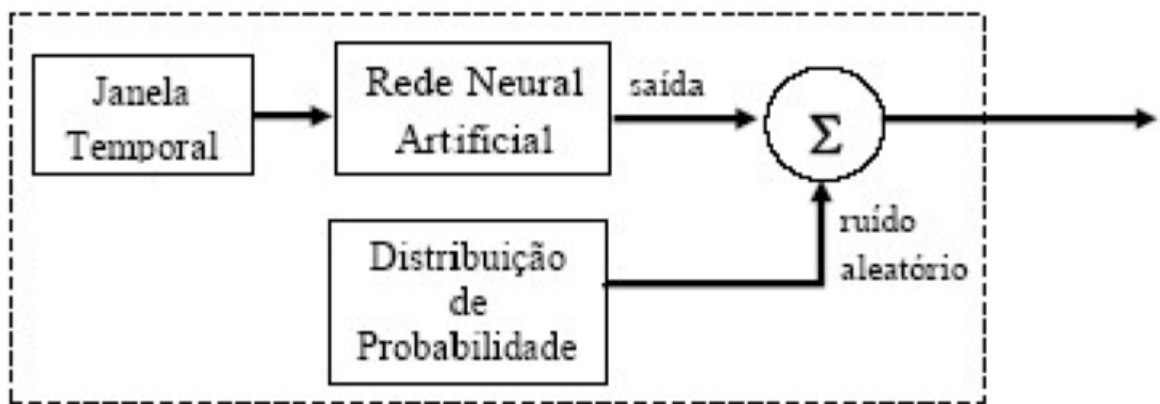


Figura 2.8: Componente Estocástica do PEN (Campos, 2010).

Como o PEN foi formado por 12 componentes estocásticas, ocorreu um encadeamento entre elas, de tal modo que o valor da série fornecido pela CE de um período estava presente na janela temporal das entradas da RNA da CE do período seguinte, conforme ilustra a Figura 2.9.

A janela temporal da RNA da CE de um período  $p$  qualquer contém uma quantidade de parâmetros da série, chamado ordem. A ordem representa os termos da auto-

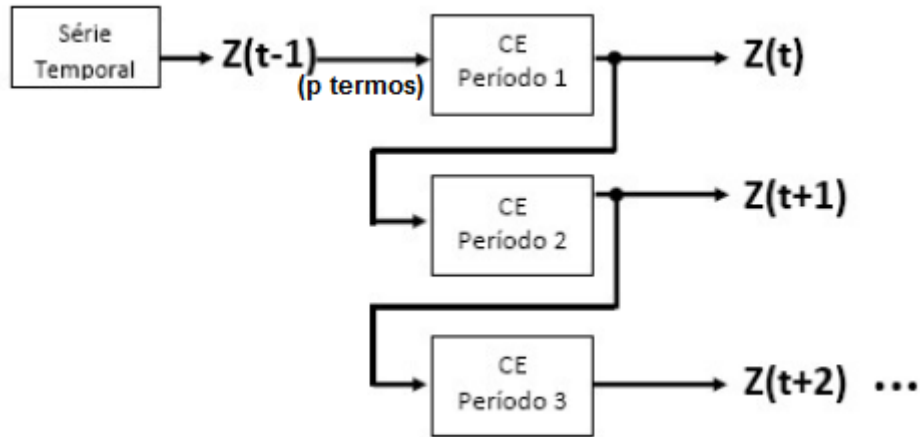


Figura 2.9: Encadeamento das CEs do PEN (Campos, 2010).

regressão, ou seja, de quantos termos a informação atual depende. Ao ser aplicado nas séries de ENA, foi decidido que a ordem poderia variar de 1 a 11, a fim de manter a sazonalidade anual. Além disso, foi acrescentado na entrada o valor da série no ano anterior para reforçar o aprendizado do comportamento mensal da série.

O histórico de ENA que foi considerado no trabalho de Campos (2010) (de 1931 a 2005) foi dividido nos conjuntos de treinamento (69 anos) e validação (5 anos), uma vez que ele era composto por somente 74 anos. Na fase de treinamento, treinou-se RNAs com ordem  $p$  ( $p = 3, 6, 9$  ou  $11$ ) e um número  $l$  de neurônios na camada escondida ( $l = 1 \dots 20$ ) para cada mês  $m$ . Após o treinamento de todas as configurações de RNA, calculou-se o Erro percentual médio absoluto (MAPE), cuja fórmula está sendo mostrada em 2.2, e escolheu-se a rede neural de cada mês  $m$  com menor MAPE para compor a  $m$ -ésima CE do PEN.

$$MAPE = \frac{1}{N} \sum_{j=1}^N \left| \frac{alvo_j - obt_j}{alvo_j} \right| * 100 \quad (2.2)$$

onde  $N$  é a quantidade de padrões de dados,  $alvo_j$  é o valor real desejado e  $obt_j$  é o valor obtido pela RNA no conjunto de validação.

Uma vez escolhida a rede, ajustou-se os resíduos gerados pela diferença entre a saída da rede e os padrões de saída do conjunto de treinamento à uma distribuição teórica de probabilidade que compôs essa CE do PEN( $p, l$ ).

Com cada modelo PEN( $p, l$ ) escolhido, gerou-se cenários de ENA e aplicou-se



---

testes de aderência e verificou-se que os cenários gerados são estatisticamente similares às séries históricas de ENA.

### 3 Metodologia

Da mesma forma que no PEN original, decidiu-se utilizar 12 CEs, onde cada uma representa um mês do ano. Cada CE possui como entradas os valores do mês que ela corresponde e dos  $p$  meses anteriores, onde  $p$  é a ordem. Para treinar cada CE, dividiu-se os dados históricos em três conjuntos: conjunto de treinamento, conjunto de validação e conjunto de teste.

Para deixar de forma mais clara, considere a componente estocástica de janeiro com ordem 11 (logo, os dados de entradas são formados pelos 11 meses anteriores e o mesmo mês no ano anterior) e os dados divididos em 64 primeiros anos para o conjunto de treinamento, 10 próximos anos para o conjunto de validação e 5 últimos anos para o de teste. O conjunto utilizado para o treinamento é formado pelos anos de 1931 até 1994, de validação pelos anos de 1995 até 2004 e teste de 2005 até 2009. Os meses de cada ano estão ordenados em ordem decrescente começando por janeiro até fevereiro. As saídas desejadas (*targets*) de cada CE do mês correspondente a ela começam no ano seguinte e vão até um ano a mais. Sendo assim, o *target* dos dados de treinamento começam em 1932 e vão até 1995, dos dados de validação são de 1996 até 2005 e do teste de 2006 até 2010. A Tabela 3.1 ilustra os dados utilizados para realizar o treino, validação e o teste da CE de janeiro com ordem 11.

Para o correto funcionamento do PEN é importante que a série histórica passe por um processo de normalização, pois através desta abordagem é possível padronizar os dados dentro de limites pré-definidos e evitar um desvio muito alto entre os dados. Os valores limites foram baseados nos limites da função de ativação da RNA. Por exemplo, uma RNA com a função de ativação do tipo tangente hiperbólica na camada escondida teria os seus dados normalizados entre -1 e 1 enquanto uma RNA com a função logística teria o valores entre 0 e 1.

Para configurar cada CE, foi feito um estudo exploratório nos parâmetros a fim de se obter a combinação que se melhor adapta a série. No início deste estudo foi necessário definir quais seriam os parâmetros e qual a faixa de valores que cada um poderia assumir

Dados de Treinamento de Janeiro					Target de Treinamento Janeiro
JAN1931	DEZ1931	NOV1931	...	FEV1931	JAN1932
JAN1932	DEZ1932	NOV1932	...	FEV1932	JAN1933
...	...	...	...	...	...
JAN1994	DEZ1994	NOV1994	...	FEV1994	JAN1995

Dados de Validação de Janeiro					Target de Validação Janeiro
JAN1995	DEZ1995	NOV1995	...	FEV1995	JAN1996
JAN2001	DEZ2001	NOV2001	...	FEV2001	JAN2002
...	...	...	...	...	...
JAN2004	DEZ2004	NOV2004	...	FEV2004	JAN2005

Dados de Teste de Janeiro					Target de Teste Janeiro
JAN2005	DEZ2005	NOV2005	...	FEV2005	JAN2006
JAN2006	DEZ2006	NOV2006	...	FEV2006	JAN2007
...	...	...	...	...	...
JAN2009	DEZ2009	NOV2009	...	FEV2009	JAN2010

Tabela 3.1: Dados da CE de janeiro com ordem 11

durante a fase de treinamento das RNAs. Na tabela 3.2 é possível visualizar quais foram esses valores.

Parâmetros da RNA	
Ordem	[1-11]
Número de Neurônios	[1-24]
Função de Ativação	[Tan. Hiperbólica, Logística]
Algoritmo de treinamento	[ <i>Levenberg-Marquardt backpropagation</i> , <i>Resilient backpropagation</i> ]

Tabela 3.2: Faixa de valores dos parâmetros das RNAs

Como mencionado, decidiu-se utilizar redes neurais *feedforward* MLP e segundo Haykin (2001), para uma RNA MLP computar aproximações de funções contínuas é necessário somente uma camada oculta, desde que se utilize funções do tipo sigmóide como função de ativação. Se baseando nisso, decidiu-se utilizar as funções tangente hiperbólica ou logística na camada escondida da RNA.

A quantidade de neurônios na camada escondida é um dos pontos que pode ser crucial para o desempenho da RNA. Mesmo que um número maior de neurônios possa dar um maior poder computacional, isso pode torná-lo mais sensível aos problemas de *overfitting*, além de aumentar o tempo de treinamento. Contrariamente, um número menor de neurônios pode não ser capaz de modelar adequadamente o problema. Ou seja,

a RNA deve ser grande o suficiente para aprender o problema e pequena o bastante para generalizar. Levando em consideração isso e que o número máximo de entradas de uma RNA pode ser 12, decidiu-se variar os neurônios na camada oculta de 1 a 24.

Para treinar as redes, utilizou-se como função de custo a função MSE e duas variações do algoritmo *backpropagation*: o algoritmo de *Levenberg-Marquardt* e o *Resilient backpropagation*. O algoritmo *backpropagation* é o mais popular e antigo algoritmo de treinamento supervisionado de MLP, proposto por Rumelhart et al (1995). Ele calcula os erros da camada de saída e repassa o erro pela rede, modificando os pesos sinápticos.

O algoritmo de *Levenberg-Marquardt* fornece uma solução numérica para o problema de minimização de uma função não-linear e possui uma convergência estável (Hagan et al, 1994). Ele se baseia, para a aceleração do treinamento, na determinação das derivadas de segunda ordem do erro em relação aos pesos e produz bons resultados para bases de dados pequenas ou médias (Kisi et al, 2005).

O algoritmo *Resilient backpropagation* é um método de aprendizagem eficiente, que realiza uma adaptação direta da etapa de peso com base nas informações locais do gradiente (Riedmiller et al, 1993). Ele usa somente o sinal da derivada para determinar a direção da atualização do peso, fazendo com que a magnitude da derivada seja desconsiderada. O tamanho da variação do peso é determinado por um valor de atualização independente. O valor de atualização de cada peso e bias é aumentado por um fator sempre que a derivada dos erros tem o mesmo sinal para duas iterações sucessivas. O valor de atualização é diminuído por um fator sempre que a derivada muda de sinal entre iterações. Se a derivada é zero, então o valor se mantém o mesmo.

Em resumo, observa-se que para cada mês  $m$  treina-se RNAs com 2 tipos diferentes de função de ativação  $\varphi$  (tangente hiperbólica ou logística) e para cada função escolhida, tem-se 2 possíveis algoritmos de treinamento  $u$  (*Levenberg-Marquardt* ou *Resilient backpropagation*), com 11 tipos de ordem diferentes  $p$  (de 1 a 11) e, para cada ordem, 24 possibilidades de neurônios  $l$  na camada oculta das RNAs, dando um total de 1056 configurações a serem avaliadas a cada mês.

O programa foi implementado utilizando o MatLab 8.1.0.604 (R2013a) (Mathworks, 2013). Os outros parâmetros para configurar a RNA foram os *defaults* desta versão, que

estão sendo apresentados na tabela 3.3.

Parâmetros das RNAs	
Taxa inicial do Aprendizado	0.01
Taxa inicial do $\mu$	0.001
Valor incremento do $\mu$	10
Valor decremento do $\mu$	0.1
Valor da constante de Momentum	0.01
Erro Mínimo	0

Tabela 3.3: Parâmetros *defaults* do MatLab para as RNAs

Para evitar que a escolha de uma configuração de RNA seja influenciada pelo mal desempenho devido a uma inicialização ruim dos pesos sinápticos, decidiu-se fazer 10 inicializações de cada configuração e obter uma série de resíduos de validação, que corresponde às diferenças entre as saídas da RNA do mês  $m$  com as saídas da desejadas desse mês  $m$  presentes no conjunto de validação.

Uma vez definida a RNA do mês  $m$  com menor MAPE calculado com os resíduos de validação, ajusta-se a série de resíduos obtidos entre a saída da RNA e os padrões de saída do conjunto de treinamento à uma distribuição teórica de probabilidade com melhor aderência a esta série. Verifica-se a aderência entre esta série e a distribuição teórica de probabilidade através do teste de Kolmogorov-Smirnov, que é um teste de aderência não paramétrico, que compara se duas distribuições de probabilidade diferem uma da outra ou se uma distribuição de probabilidade difere da distribuição em hipótese. O teste é baseado na maior diferença absoluta entre uma função de distribuição acumulada empírica e outra hipotética (Massey et al, 1951). A distribuição teórica de probabilidade que conseguir o menor erro de estimativa através deste teste é adotada como a distribuição de probabilidade dos resíduos que compõem a CE do mês  $m$  no  $PEN(\varphi, u, p, l)$ . Definindo-se essa função de distribuição de probabilidade, utiliza-se o modelo  $PEN(\varphi, u, p, l)$  para gerar séries sintéticas de ENA.

A geração de séries sintéticas é feita de forma sequencial e a cada instante de tempo a série é obtida como uma função de seus valores passados e de um termo aleatório. Esse valor aleatório é gerado através de um sorteio que utiliza o método de amostragem de Quase Monte Carlo com sequência de Sobol, visando capturar os elementos uniformemente distribuídos dentro da distribuição de probabilidade (Sobol, 1967). Os valores iniciais do

processo de geração de séries sintéticas são tirados diretamente da série histórica original.

Um pseudo-código que representa todo o processo é mostrado na Figura 3.1.

```

1 Para cada regioao reg = [SE,S,N,NO]
2   % Normaliza a série
3   serieN = normalizaSerieHistorica(serieRegiao);
4   % Obtém os conjuntos de treinamento (ct), validação (cv) e teste.
5   [ct, cv, ct] = obtemConjuntosSerie(serieN);
6   Para cada mês m = [1:12]
7     Para cada f = [tgh, log]
8       Para cada u = [LM, RB]
9         Para cada p = [1:11]
10          Para cada l = [1:24]
11            Para cada inicializacao i = [1:10]
12              % Cria uma RNA utilizando a função de ativação f na camada
13              % escondida, algortimo de treinamento u, ordem p e
14              % nº l de neurônios na camadada escondida
15              rede = configuraRedeCEPEN(f, u, p, l);
16              treinaRede(rede, ct);
17              mape = validaRede(rede, cv)
18              Se mape é o menor
19                melhorInicializacao = rede
20              fimSe;
21            fimPara;
22            % Salva a inicialização do PEN(f,u,p,l) que teve menor MAPE
23            salvaRedeCEPEN(melhorInicializacao);
24          fimPara;
25          Para cada rede r salva
26            mape = validaRede(r, cv)
27            Se mape é o menor
28              melhorCE = r;
29            fimSe;
30          fimPara;
31          dist = obtemDistribuicaoProbabilidade(melhorCE, ct)
32          salvaCEPEN(melhorCE, dist, m, f, u, p)
33        fimPara;
34      fimPara;
35    fimPara;
36  fimPara;
37  % Para cada uma das 44 melhores configurações do modelo PEN(f,u,p,l)
38  Para cada melhor cofiguracao conf = [1:44]
39    % Obtém a função de distribuição
40    distProbabilidade = CEPEN(conf).distribuicaoProbabilidade;
41    % Gera os cenário
42    Para cada cenario c = [1:200]
43      Para cada mês m = [1:60]
44        valor = obtemValorMes(m);
45        ruido = obtemValorDistribuicaoProbabilidade(distProbabilidade);
46        cenarios(c,m) = valor + ruido;
47      fimPara;
48    fimPara;
49    salvaCenarios(cenarios);
50  fimPara;
51 fimPara;

```

Figura 3.1: Pseudo-código do treinamento e da geração dos cenários

## 4 Resultados

### 4.1 Análise do ajuste do novo modelo PEN

Para definir a quantidade de anos para os conjuntos de validação e teste, variou-se os anos de validação e teste de forma empírica e constatou-se que a melhor combinação é 5 anos para teste e 10 para validação.

Tendo definido o tamanho dos conjuntos de treinamento, validação e teste, realizou-se os testes descritos no Capítulo 3. Conforme descrito neste capítulo, decidiu-se testar para cada mês 2 tipos diferentes de função de ativação  $\varphi$ , 2 possíveis algoritmos de treinamento  $u$ , 11 tipos de ordem  $p$  e 24 possibilidades de neurônios  $l$  na camada oculta das RNAs. Logo, para cada mês, treinou-se 1056 configurações de RNAs para cada subsistema do SIN e além disso, cada uma dessas configurações foram treinadas 10 vezes, com diferentes inicializações dos pesos sinápticos.

Na fase de treinamento gerou-se um arquivo com o valor do MAPE da melhor inicialização de cada configuração de CE. Após o treinamento de todas as configurações, analisou-se os resultados e escolheu-se as 44 melhores configurações de redes neurais de cada CE de cada subsistema. Para fazer esta escolha, levou-se em consideração o menor MAPE das 24 possibilidades de neurônios na camada oculta das RNAs para cada ordem, algoritmos de treinamento e função de ativação. Desta forma, os parâmetros  $\varphi, u$  e  $p$  do modelo  $PEN(\varphi, u, p, l)$  vão ser os mesmos para cada uma de suas CE e o parâmetro  $l$  pode variar de uma para a outra. A Tabela 4.1 representa um exemplo de uma das configurações escolhidas. Ela apresenta o MAPE e o número de neurônios na camada escondida de cada CE da configuração do  $PEN('log', 'lm', 1, l)$  do subsistema SE/CO.

Treinamento: <i>Levenberg-Marquardt</i>					Função de ativação: Logística						Ordem: 1	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
<b>MAPE</b>	8,08	8,87	4,91	11,77	3,23	4,91	8,40	7,53	16,07	22,98	9,54	12,77
$l$	3	14	5	24	10	15	1	23	20	20	22	17

Tabela 4.1:  $PEN('l', 'lm', 1, l)$  do subsistema Sudeste/Centro-Oeste

## 4.2 Análise dos cenários sintéticos gerados

Tendo cada configuração de RNA escolhida, calculou-se a série de resíduos de treinamento e a partir dela, encontrou-se a distribuição de probabilidade teórica com a melhor aderência. As distribuições testadas foram a distribuição Logística, Loglogística, Lognormal e Normal. Após determinar a distribuição de probabilidade e a configuração da RNA de cada CE do modelo  $PEN(\varphi, u, p, l)$  é possível utilizá-lo para gerar cenários sintéticos de ENA.

Conforme o modelo PEN original, foram gerados 200 cenários de 60 meses de ENA de cada subsistema. O ano base utilizado para a geração dos cenários foi o primeiro ano do conjunto de teste (ano 2005).

Para verificar a aderência dos cenários gerados aplicou-se os mesmos testes de aderência do modelo PEN original: o teste  $t$ , o teste de Levene e o teste de Kolmogorov-Smirnov (K-S). O teste  $t$  é uma análise estatística de duas médias populacionais. Neste trabalho, utilizou-se ele para verificar se a média é estatisticamente igual à média do histórico. Para isso, calculou-se a média dos 200 valores de cada mês em cada ano dos cenários e os comparou com o valor do histórico daquele mês e ano, obtendo um  $p$ -valor. Por convenção, se houver menos de 5% de chance de conseguir esse valor ao acaso, então rejeita-se a hipótese do teste: a média do cenário de um mês não é estatisticamente igual à média histórica desse mês. Esse teste é aplicado para todos os 60 meses, gerando 60  $p$ -valores. Para avaliar o desempenho dos cenários gerados, calculou-se a porcentagem de  $t$ -valores maiores que 5%. O teste de Levene é aplicado para avaliar a homogeneidade de variância. Ele foi utilizado para verificar se a variância de cada período dos cenários é estatisticamente igual à variância do histórico do mês correspondente. O teste de K-S foi utilizado para analisar se os cenários provêm da mesma distribuição de probabilidade do histórico.

A Tabela 4.2 apresenta a melhor configuração de cada algoritmo em cada subsistema. Observe que todos os subsistemas obtiveram uma boa aderência com o conjunto histórico de teste (2005 até 2009). Além disso, os valores obtidos foram, na sua maioria, maiores ou iguais aos valores obtidos no PEN original, indicando que as alterações feitas no modelo PEN geraram um modelo tão bom quanto o original.



Subsistema	Treinamento	Função de ativação	$t$	Levene	K-S
SE/CO	<i>Levenberg-Marquardt</i>	Tangente hiperbólica	95	90	99
	<i>Resilient backpropagation</i>	Logística	99	100	90
	PEN Original		94	95	94
S	<i>Levenberg-Marquardt</i>	Logística	95	90	99
	<i>Resilient backpropagation</i>	Logística	90	85	85
	PEN Original		90	90	75
N	<i>Levenberg-Marquardt</i>	Logística	94	95	97
	<i>Resilient backpropagation</i>	Logística	94	100	87
	PEN Original		85	90	64
NE	<i>Levenberg-Marquardt</i>	Logística	95	94	90
	<i>Resilient backpropagation</i>	Logística	97	93	99
	PEN Original		97	99	94

Tabela 4.2: Comparação dos testes de aderência dos cenários no melhores cenários de cada algoritmo de treinamento gerados no  $PEN(\varphi, u, p, l)$  e no PEN original

A Tabela 4.3 apresenta a estrutura do  $PEN(\varphi, u, p, l)$  da melhor configuração de cada algoritmo em cada subsistema. A distribuição de probabilidade (D. P.) Logística foi abreviada para ‘LO’, a distribuição Loglogística para ‘LL’, a distribuição Lognormal para ‘LN’ e a distribuição Normal para ‘NO’.

Como é possível ver, em quase todos os casos a melhor função de ativação foi a Logística (exceto no  $PEN(\varphi, u, p, l)$  com algoritmo de treinamento *Levenberg-Marquardt* do Subsistema SE/CO). Além disso, as ordens dos melhores resultados dos subsistemas SE/CO e S são pequenas, indicando que foi necessário pouca informação para a rede aprender o comportamento da série, enquanto as ordens dos subsistemas N e NE são elevadas.

Os gráficos da Tabela 4.4 apresentam a curva da média de cada mês do conjunto de teste (ano de 2005 até 2009) replicada nos 5 anos, os 200 cenários gerados de 60 meses de ENA e a curva da média de cada mês dos 200 cenários da melhor configuração de cada algoritmo em cada subsistema. Desta maneira é possível analisar o comportamento da média dos cenários nos 5 anos.

Subsistema Sudeste/Centro-Oeste												
Treinamento: <i>Levenberg-Marquardt</i>				Função de ativação: Tangente hiperbólica							Ordem: 1	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	NO	NO	LL	LN	LN	NO	LL	LN	NO	LO	NO	NO
<i>l</i>	22	5	12	19	23	4	21	24	4	17	11	10
Treinamento: <i>Resilient backpropagation</i>				Função de ativação: Logística							Ordem: 4	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	LL	LN	NO	LN	LL	LO	NO	LN	NO	LO	NO	LL
<i>l</i>	16	18	11	24	5	21	3	14	13	19	12	18

Subsistema Sul												
Treinamento: <i>Levenberg-Marquardt</i>				Função de ativação: Logística							Ordem: 2	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	LN	NO	LL	LN	LN	NO	NO	LN	LL	LO	NO	NO
<i>l</i>	3	8	18	23	17	11	22	6	5	16	7	18
Treinamento: <i>Resilient backpropagation</i>				Função de ativação: Logística							Ordem: 3	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	LL	NO	LO	LL	LO	NO	LL	NO	NO	LO	NO	LO
<i>l</i>	3	23	23	11	20	15	7	20	7	14	18	2

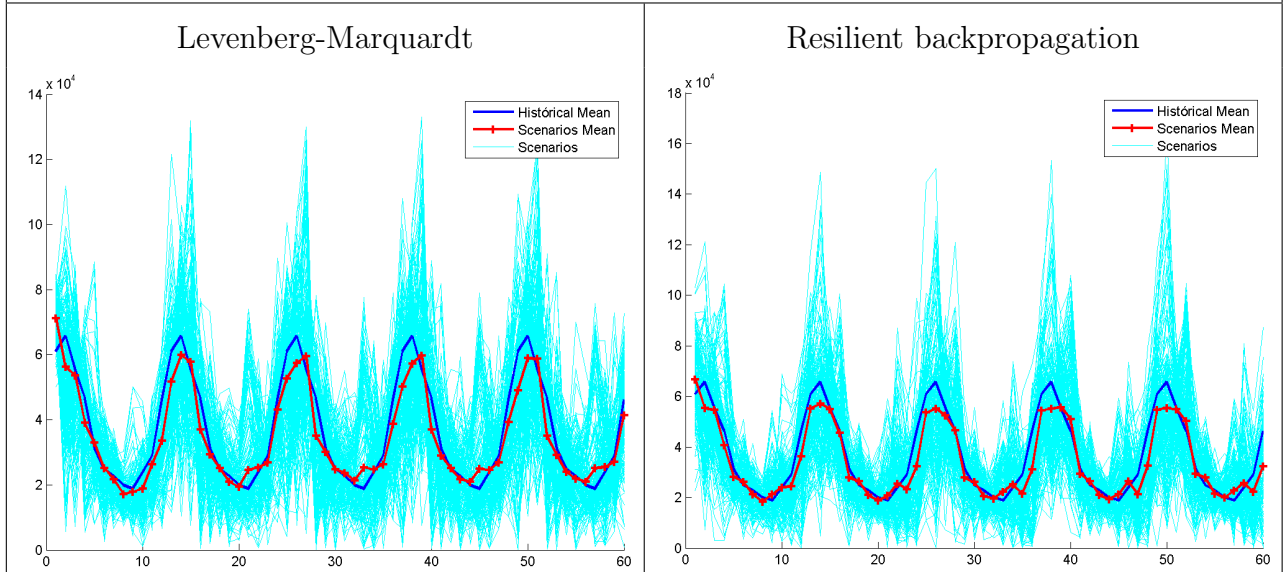
Subsistema Norte												
Treinamento: <i>Levenberg-Marquardt</i>				Função de ativação: Logística							Ordem: 10	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	LN	LL	LN	LN	LL	NO	NO	LN	LL	LO	LL	LL
<i>l</i>	2	11	3	15	13	13	3	16	12	11	12	19
Treinamento: <i>Resilient backpropagation</i>				Função de ativação: Logística							Ordem: 10	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	LN	NO	LN	LN	LO	NO	LL	LL	NO	LO	LL	LO
<i>l</i>	8	18	20	4	15	23	4	22	9	22	22	8

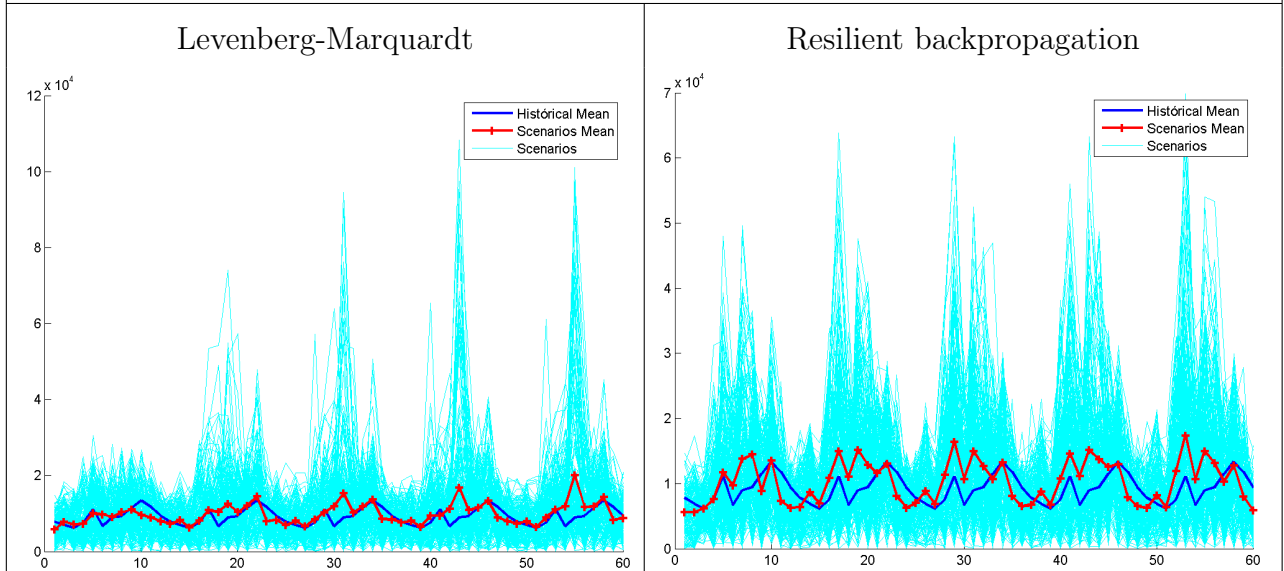
Subsistema Nordeste												
Treinamento: <i>Levenberg-Marquardt</i>				Função de ativação: Logística							Ordem: 10	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	NO	NO	LL	LN	LN	LN	LL	LN	LN	LL	LL	LL
<i>l</i>	22	17	2	19	6	12	22	11	15	16	1	21
Treinamento: <i>Resilient backpropagation</i>				Função de ativação: Logística							Ordem: 10	
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
D. P.	LL	NO	LN	NO	LN	LN	LN	LL	NO	LL	LO	LO
<i>l</i>	10	8	13	3	18	12	2	7	19	10	18	7

Tabela 4.3: Estrutura dos Melhores  $PEN(\varphi, u, p, l)$

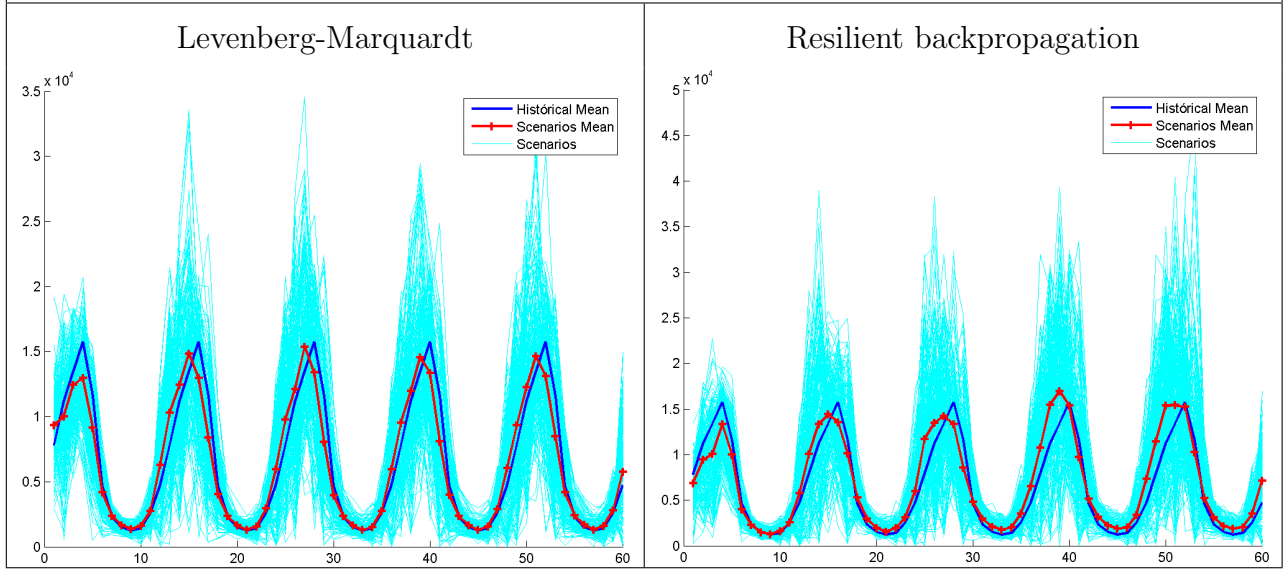
**Subsistema Sudeste/Centro-Oeste**



**Subsistema Sul**



**Subsistema Norte**



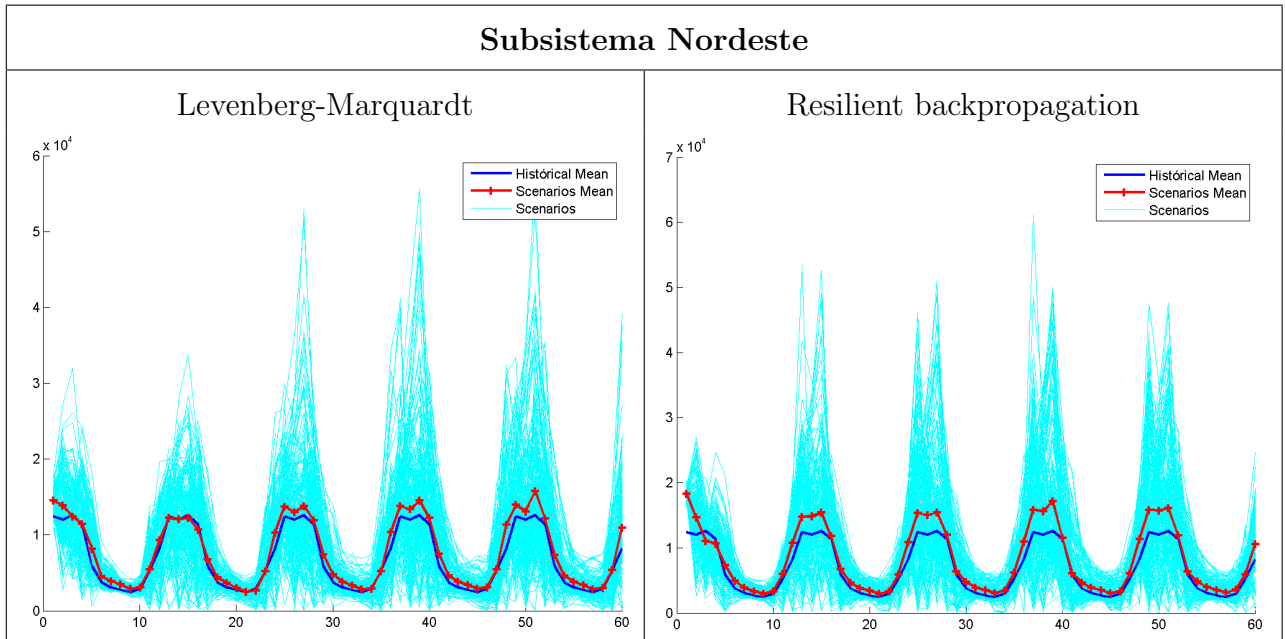


Tabela 4.4: Envoltória dos cenários gerados

Observa-se que a média dos conjuntos históricos ficaram dentro das envoltórias de cenários em todos os subsistemas. Além disso, a média dos cenários da configuração com o treinamento *Levenberg-Marquardt* acompanha melhor a média dos conjuntos históricos do que a configuração com o treinamento *Resilient backpropagation* nos 4 subsistemas, indicando que o algoritmo de treinamento *Levenberg-Marquardt* resulta em uma rede mais adaptada à série histórica. No subsistema SE/CO, as médias dos cenários dos meses de março e de maio a agosto são as que melhor acompanham a média do conjunto histórico. Nos subsistemas N e NE, as médias dos cenários nos meses de janeiro a abril não acompanham muito a média do conjunto histórico. No subsistema S, não ocorreu um período longo em que a média dos cenários acompanham com precisão a média do conjunto histórico.

Resumindo a análise dos modelos  $PEN(\varphi, u, p, l)$ , concluiu-se que o modelo PEN com algoritmo de treinamento *Levenberg-Marquardt* foi o que gerou cenários mais aderentes às séries históricas em todos os subsistemas. Para o subsistema SE/CO, esse modelo utiliza a função de ativação tangente hiperbólica na camada oculta. Para os outros subsistemas, a função de ativação é a logística. As ordens para os subsistemas SE/CO, S, N e NO são 1, 2, 10 e 10, respectivamente.

## 5 Considerações finais

A série histórica de ENA possui dados insuficientes para compor uma amostra de tamanho necessário para estimar riscos aceitáveis no planejamento da operação do sistema hidrotérmico brasileiro. Uma maneira de avaliar estes riscos é a geração de cenários estatisticamente aderentes a série. Conforme mostrado por Campos (2010), o modelo PEN gera cenários com uma boa aderência a série histórica.

Este trabalho apresenta um estudo sobre modificações no modelo PEN e testa-lo no tratamento das incertezas das séries de ENA. Como foi visto no capítulo 4, testou-se várias configurações do modelo  $PEN(\varphi, u, p, l)$ , variando a função de ativação  $\varphi$ , o algoritmo de treinamento  $u$ , a ordem  $p$  e o número  $l$  de neurônios na camada escondidas das RNAs de cada CE do PEN. As melhores configurações foram escolhidas para a geração de 200 cenários de 5 anos de ENA. Em cada cenário foram aplicados 3 testes de aderência e foi verificado que os cenários gerados estavam estatisticamente similares às séries históricas tão quanto os cenário gerado pelo PEN proposto por Campos (2010). Além disso, foi verificado que as RNAs treinadas com o algoritmo *Levenberg-Marquardt* tiveram uma melhor adaptação à série histórica em todos os subsistemas. Segundo os resultados obtidos aqui, no subsistema Sudeste/Centro-Oeste, a melhor configuração usa como função de ativação na camada oculta a função tangente hiperbólica e tem ordem 1. Já para o subsistema Sul, a melhor função de ativação na camada oculta é a logística e ordem 2. Nos outros subsistemas, a melhor função de ativação na camada oculta é a logística e ordem 10.

Como propostas futuras, é possível que se possa obter melhores resultados testando diferentes parâmetros para os algoritmos de treinamento e para as funções de ativação. Além disso, pode ser possível conseguir cenários ainda mais aderentes refinando os ruídos provenientes das função distribuição de probabilidade. Uma outra proposta consiste em utilizar redes recorrentes.

## Referências Bibliográficas

- ANEEL. **Bandeiras tarifárias**. Disponível em: <http://www.aneel.gov.br/area.cfm?idArea=758>. Acesso em: 02 maio 2014.
- ANEEL. **Big - banco de informações de geração**. Disponível em: <http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>. Acesso em: 02 maio 2014.
- Ballini, R.; França, E.; Kadowaki, M.; Soares, S. ; Andrade, M. Modelos de redes neurais e box & jenkins para previsão de vazões médias mensais. **Simpósio Brasileiro de Recursos Hídricos, 12º. CD-ROM**, 1997.
- Ballini, R.; Soares, S. ; Andrade, M. Previsão de vazões médias mensais usando redes neurais nebulosas. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, v.14, n.3, p. 680–693, 2003.
- Batista, A. L. F. **Modelos de séries temporais e redes neurais artificiais na previsão de vazão**. Lavras, MG, 2009. Metrado em engenharia de sistemas - Universidade Federal de Lavras.
- Campos, L. C. D. **Um Modelo Estocástico baseado em Redes Neurais**. Rio de Janeiro, RJ, 2010. Doutorado em engenharia elétrica - Pontifícia Universidade Católica do Rio de Janeiro.
- dos Santos Coelho, L.; Junior, O. Rede neural de base radial aplicada em previsão de séries temporais: algoritmo e aplicação. **XX Encontro Nacional de Engenharia de Produção**, 2000.
- Elman, J. L. Finding structure in time. **Cognitive Science**, v.14, p. 179–211, 1990.
- Fiorin, D. V.; Martins, F. R.; Schuch, N. J. ; Pereira, E. B. Forecast of solar energy resource by using neural network methods. **Revista Brasileira de Ensino de Física**, v.33, n.1, p. 01–20, 2011.
- Hagan, M. T.; Menhaj, M. B. Training feedforward networks with the marquardt algorithm. **Neural Networks, IEEE Transactions on**, v.5, n.6, p. 989–993, Novembro 1994.
- Haykin, S. **Redes neurais-princípios e prática**. Porto Alegre, RS: Bookman, 2001.
- ONS. **Interligação acre-rondônia gera economia de r\$ 147 milhões**. Disponível em: <http://www.ons.org.br/newsletters/informativos/jan2010/07-materia01.html>. Acesso em: 02 maio 2014.
- Kisi, Ö.; Uncuoglu, E. Comparison of three back-propagation training algorithms for two case studies. **Indian journal of engineering & materials sciences**, v.12, n.5, p. 434–442, 2005.
- Kovács, Z. **Redes neurais artificiais: fundamentos e aplicações**. Editora Livraria da Física, 2002.

- Massey Jr, F. J. The kolmogorov-smirnov test for goodness of fit. **Journal of the American statistical Association**, v.46, n.253, p. 68–78, 1951.
- Mathworks. **The mathworks - matlab: The language of technical computing**. <http://www.mathworks.com>.
- Morettin, P. A.; Toloi, C. **Análise de séries temporais**. Blucher, 2006.
- Müller, D. **Processos estocásticos e aplicações**. In: Coleção Econômicas n.º, 2ª Série de Direito Financeiro e Tributário. Almedina, 2007.
- Papoulis, A.; Pillai, S. **Probability, random variables and stochastic processes with errata sheet**. New York: McGraw-Hill Science/Engineering/Math, 2001.
- Riedmiller, M.; Braun, H. **A direct adaptive method for faster backpropagation learning: The rprop algorithm**. In: Neural Networks, 1993., IEEE International Conference on, p. 586–591. IEEE, 1993.
- Rumelhart, D. E.; Durbin, R.; Golden, R. ; Chauvin, Y. Backpropagation: The basic theory. **Backpropagation: Theory, Architectures and Applications**, p. 1–34, 1995.
- Sobol, I. M. On the distribution of points in a cube and the approximate evaluation of integrals. **USSR Computational Mathematics and Mathematical Physics**, v.7, n.4, p. 86–112, 1967.
- Souza, R. C.; CAMARGO, M. E. Análise e previsão de séries temporais: os modelos arima. **Ijuí, RS: Sedigraf**, v.242, 2004.
- Sousa, W. d. S.; Sousa, F. As rede neural artificial aplicada à previsão de vazão da bacia hidrográfica do rio piacó. **Revista Brasileira de Engenharia Agrícola e Ambiental**, v.14, n.2, p. 173–180, 2010.
- Wasserman, P. **Neural computing: theory and practice**. Van Nostrand Reinhold Co., 1989. il.

# A Resultados adicionais

## A.1 Melhores CE de cada subsistema

Os resultados aqui apresentados são referentes as melhores CE da 44 melhores configurações do  $PEN(\varphi, u, p, l)$  de cada subsistema do SIN. O algoritmo *Levenberg-Marquardt* está abreviado como *LM* e o algoritmo *Resilient backpropagation* como *RB*. Os nomes das funções de ativação também estão abreviados: *Tgh* para a função tangente hiperbólica e *Log* para a logística.



Subsistema Sudeste/Centro-Oeste												
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
MAPE	3,89	1,91	2,14	3,33	1,55	0,73	1,87	1,92	5,25	2,90	1,55	4,81
<i>l</i>	18	7	8	15	17	17	11	10	16	10	10	6
Ordem	8	5	2	11	3	7	11	11	11	2	11	11
Fun. At.	<i>Log</i>	<i>Log</i>	<i>Log</i>	<i>Tgh</i>	<i>Log</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Log</i>	<i>Log</i>
Alg. Tr.	<i>RB</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>RB</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>RB</i>	<i>RB</i>

Subsistema Sul												
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
MAPE	0,45	5,11	3,98	8,00	7,68	9,88	2,37	5,02	11,41	11,60	5,98	5,02
<i>l</i>	15	3	22	24	22	20	8	12	1	17	18	13
Ordem	10	10	9	9	10	2	3	3	11	11	9	1
Fun. A.	<i>Log</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>
Alg. Tr.	<i>LM</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>LM</i>	<i>LM</i>	<i>LM</i>

Subsistema Norte												
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
MAPE	2,98	3,14	2,02	1,43	9,12	2,02	2,86	0,96	2,42	5,41	9,38	2,31
<i>l</i>	21	8	23	17	15	22	12	14	16	24	5	17
Ordem	1	1	8	4	6	10	11	8	9	11	11	6
Fun. A.	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Log</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Log</i>	<i>Tgh</i>	<i>Log</i>	<i>Tgh</i>
Alg. Tr.	<i>RB</i>	<i>RB</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>	<i>RB</i>	<i>LM</i>

Subsistema Nordeste												
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
MAPE	4,01	5,90	2,37	3,58	2,26	1,30	1,11	1,28	2,11	4,52	11,75	8,29
<i>l</i>	23	17	4	11	6	15	6	21	19	22	6	14
Ordem	4	4	5	3	7	8	10	10	11	9	11	6
Fun. A.	<i>Log</i>	<i>Tgh</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Tgh</i>	<i>Log</i>	<i>Tgh</i>	<i>Tgh</i>	<i>Log</i>	<i>Tgh</i>
Alg. Tr.	<i>LM</i>	<i>RB</i>	<i>RB</i>	<i>LM</i>	<i>LM</i>	<i>LM</i>	<i>LM</i>	<i>LM</i>	<i>LM</i>	<i>RP</i>	<i>LM</i>	<i>LM</i>

Tabela A.1: Melhores CEs de cada subsistema do SIN