

Universidade Federal do Pará
Instituto de Tecnologia
Programa de Pós-Graduação de Engenharia Elétrica

Flaviano Ramos Pereira Junior

Redes neurais diretas e recorrentes na previsão do preço de energia elétrica
de curto prazo no mercado brasileiro

Dissertação de Mestrado

Volume I

Belém
11 de Novembro de 2016

Flaviano Ramos Pereira Junior

**Redes neurais diretas e recorrentes na previsão do preço de energia elétrica
de curto prazo no mercado brasileiro**

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica, na área de Computação Aplicada.

Orientador: Roberto Célio Limão de Oliveira

Volume I

Belém

11 de Novembro de 2016

Dados Internacionais de Catalogação - na – Publicação (CIP) Sistema de Bibliotecas da UFPA

Pereira Júnior, Flaviano Ramos, 1988-

Redes neurais diretas e recorrentes na previsão do preço de energia elétrica de curto prazo no mercado brasileiro. / Flaviano Ramos Pereira Júnior.. - 2016.

Orientador: Roberto Célio Limão de Oliveira

Dissertação (mestrado) - Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica , Belém, 2016.

1. Redes neurais (computação). 2. Serviços de eletricidade – custos . 3. Energia elétrica – aspectos econômicos. I. Título.

CDD 23. ed. 006.32

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS GRADUAÇÃO EM ENERGIA ELÉTRICA

**“REDES NEURAIIS DIRETAS E RECORRENTES NA PREVISÃO DE
ENERGIA ELÉTRICA DE CURTO PRAZO NO MERCADO BRASILEIRO”**

AUTOR: FLAVIANO RAMOS PEREIRA JUNIOR

DISSERTAÇÃO DE MESTRADO APRESENTADA AO PROGRAMA DE PÓS
GRADUAÇÃO EM ENGENHARIA ELÉTRICA, COMO PARTE DOS REQUISITOS
NECESSÁRIO À OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA
ELÉTRICA, NA ÁREA DE COMPUTAÇÃO APLICADA

APROVADA EM 11/09/2016

BANCA EXAMINADORA:

Prof. Dr. Roberto Célio Limão
(Orientador – PPGEE/UFPA)

Profa. Dra. Adriana Rosa Garcez de Castro
(Avaliadora interna - PPGEE/UFPA)

Prof. Dr. Jefferson Magalhães de Moraes
(avaliador externo ao programa – PPGCC/UFPA)

Prof. Dr. José Carlos Reston Filho
(avaliador externo ao programa – IDAAM - Manaus)

VISTO:

Prof. Dr. Evaldo Gonçalves Pelaes
(Coordenador do PPGEE/ITEC/UFPA)

Dedico este trabalho à minha família.

Agradecimentos

Agradeço aos meus pais, Flaviano Ramos Pereira Neto e Rossana Patrícia Souza de Almeida, pelo amor, educação e por sempre incentivarem e proporcionarem as melhores condições para os meus estudos. Ao meu irmão Flávio Guarany Ramos Pereira Neto pelo companheirismo em todos os momentos.

Agradeço ao meu orientador Roberto Célio Limão de Oliveira pela oportunidade, incentivo e direcionamento a este trabalho.

Agradeço à minha namorada Adriene Cardoso por estar ao meu lado em todos os momentos incentivando esta jornada.

“O que o homem não conhece não existe para ele. Por isso, o mundo tem, para cada um, o tamanho que abrange seu conhecimento” - Albert Einstein

Resumo

Nos estudos sobre o mercado de energia do Brasil existem poucos trabalhos sobre predição do preço de energia elétrica em curto prazo. Os que existem utilizam modelos preditores do tipo ARIMA e rede neural direta, entretanto com a rede neural sem método de seleção das variáveis de entrada ou dos atrasos das entradas. Além disso, não há trabalhos que utilizem redes neurais recorrentes no mercado brasileiro. O mercado de energia de curto prazo pode apresentar importantes oportunidades aos agentes atuantes, pois a comercialização nesse mercado é menos burocrática em relação ao mercado de longo prazo. Este trabalho apresenta o uso de redes neurais diretas e recorrentes (além da comparação com o modelo ARIMA) para a previsão do preço de energia elétrica de curto prazo brasileiro com uso da técnica de correlação para seleção das variáveis externas da rede e também para escolha dos atrasos nestas variáveis selecionadas. Mostra-se que, na previsão de um passo a frente, as redes neurais implementadas superam o desempenho do modelo ARIMA para esta série e, em geral, a rede direta apresenta melhor resultado que a recorrente. Além disso, a seleção dos atrasos nas variáveis de entrada melhora o desempenho da rede neural direta.

Abstract

There are few articles about short term electricity price prediction in the Brazilian market. Existing works use ARIMA predictors and feedforward neural networks however, without input selection or lag selection for these inputs. Besides, there is no work with use of recurrent neural networks in the Brazilian electricity market. The short term electricity market may show important opportunities for the agents acting as the commercialization in this market is less bureaucratic in relation to the long-term market.. This article shows the use of feedforward and recurrent neural networks (besides comparison with the ARIMA model) to predict short term electricity price with the use of correlation for exogenous input selection for the networks and also for lag selection to these inputs. It is shown that, for one step forward predictions, both implemented networks outperforms the ARIMA model, and in general, feedforward network works better than recurrent network. Besides, lag selection in the input improves feedforward network performance.

Lista de ilustrações

Figura 1 – Instituições do setor elétrico brasileiro.	22
Figura 2 – Estrutura da capacidade instalada do SIN por fonte de energia.	23
Figura 3 – Submercados de energia no brasil.	24
Figura 4 – Processo de decisão em um sistema hidrotérmico.	25
Figura 5 – Custos de geração imediato e futuro e decisão ótima.	26
Figura 6 – PLD no mercado sudeste/centro-oeste	29
Figura 7 – Estrutura do neurônio biológico	31
Figura 8 – Modelo de nerurônio artificial	32
Figura 9 – Exemplo de RNA direta	33
Figura 10 – Abordagem conservadora	35
Figura 11 – Abordagem liberal	36
Figura 12 – Abordagem radical	37
Figura 13 – Treinamento supervisionado	38
Figura 14 – Arquitetura da rede neural recorrente	42
Figura 15 – Erro de validação ideal x erro de treino ao longo das épocas	51
Figura 16 – Erro de validação real	52
Figura 17 – Seleção de lags: Variável externa Carga, no submercado Sul, patamar de carga pesado.	57
Figura 18 – Seleção de lags: Variável externa geração hidráulica, submercado Nordeste, patamar de carga médio.	57
Figura 19 – Utilização dos preditores na previsão do PLD	61
Figura 20 – Erro percentual médio absoluto no submercado Sudeste/Centro-Oeste	64
Figura 21 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sudeste/Centro-Oeste com patamar de carga leve.	64
Figura 22 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sudeste/Centro-Oeste com patamar de carga médio.	65
Figura 23 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sudeste/Centro-Oeste com patamar de carga pesado.	66
Figura 24 – Erro percentual médio absoluto no submercado Sul	68
Figura 25 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sul com patamar de carga leve.	68
Figura 26 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sul com patamar de carga médio.	69
Figura 27 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sul com patamar de carga pesado.	70
Figura 28 – Erro percentual médio absoluto no submercado Nordeste	72

Figura 29 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Nordeste com patamar de carga leve.	72
Figura 30 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Nordeste com patamar de carga médio.	73
Figura 31 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Nordeste com patamar de carga pesado.	74
Figura 32 – Erro percentual médio absoluto no submercado Norte	76
Figura 33 – PLD real x PLD previsto pela rede recorrente. Submercado Norte com patamar de carga leve.	76
Figura 34 – PLD real x PLD previsto pela rede recorrente. Submercado Norte com patamar de carga médio.	77
Figura 35 – PLD real x PLD previsto pela rede recorrente. Submercado Norte com patamar de carga pesado.	78

Lista de tabelas

Tabela 1 – Correlação para o mercado Sudeste/Centro-Oeste	55
Tabela 2 – Correlação para o submercado Norte	55
Tabela 3 – Correlação para o submercado Nordeste	56
Tabela 4 – Correlação para o submercado Sul	56
Tabela 5 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Sudeste/Centro-Oeste	62
Tabela 6 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Sudeste/Centro-Oeste	63
Tabela 7 – Erro médio quadrático normalizado utilizando rede direta sem seleção de lags no submercado Sudeste/Centro-Oeste	63
Tabela 8 – Erro quadrático médio normalizado utilizando rede neural direta com seleção de lags no submercado Sudeste/Centro-Oeste	63
Tabela 9 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Sul	66
Tabela 10 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Sul	66
Tabela 11 – Erro quadrático médio normalizado utilizando rede direta sem seleção de lags no submercado Sul	67
Tabela 12 – Erro quadrático médio normalizado utilizando rede direta com seleção de lags no submercado Sul	67
Tabela 13 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Nordeste	70
Tabela 14 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Nordeste	70
Tabela 15 – Erro quadrático médio normalizado utilizando rede direta sem seleção de lags no submercado Nordeste	71
Tabela 16 – Erro quadrático médio normalizado utilizando rede direta com seleção de lags no submercado Nordeste	71
Tabela 17 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Norte	74
Tabela 18 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Norte	74
Tabela 19 – Erro quadrático médio normalizado utilizando rede direta sem seleção de lags no submercado Norte	75
Tabela 20 – Erro quadrático médio normalizado utilizando rede direta com seleção de lags no submercado Norte	75

Lista de abreviaturas e siglas

ACL	Ambiente de contratação livre
ACR	Ambiente de contratação regulada
ANEEL	Agência nacional de energia elétrica
ARFIMA	Auto-regressive fractionally integrated moving average
ARIMA	Auto-regressive integrated moving average
CCEE	Câmara de comercialização de energia elétrica
CMO	Custo marginal de operação
CMSE	Comitê de monitoramento do setor elétrico
DCT	Discrete cosine transforms
ELM	Extreme learning machine
EPE	Empresa de pesquisa elétrica
FCF	Função de custo futuro
FCI	Função de custo imediato
MAE	Mercado atacadista de energia
MLP	Multi-layer perceptron
MME	Ministério de minas e energia
ONS	Operador nacional do sistema elétrico
PCA	Principal component analysis
PLD	Preço de liquidação de diferenças
RNA	Rede neural artificial
SIN	Sistema integrado nacional

Sumário

1	INTRODUÇÃO	15
1.1	Motivação	15
1.2	Objetivos	16
1.3	Revisão bibliográfica	17
1.4	Estrutura do trabalho	18
2	COMERCIALIZAÇÃO DE ENERGIA ELÉTRICA NO BRA- SIL	20
2.1	Contexto Histórico	20
2.2	Novo marco regulatório	21
2.3	O sistema interligado nacional	23
2.4	Ambientes de contratação	24
2.5	Modelo de despacho centralizado (<i>tight pool</i>)	25
2.6	Mercado de curto prazo	27
2.7	Preço de liquidação de diferenças (PLD)	27
3	REDES NEURAIS ARTIFICIAIS	30
3.1	Neurônio biológico x neurônio artificial	31
3.2	Tipos de redes neurais	32
3.2.1	Redes neurais diretas	32
3.2.2	Redes neurais recorrentes	33
3.2.2.1	Arquiteturas de redes neurais recorrentes	34
3.2.2.1.1	<i>Abordagem conservadora</i>	34
3.2.2.1.2	<i>Abordagem liberal</i>	35
3.2.2.1.3	<i>Abordagem radical</i>	36
3.3	Treinamento de redes neurais	37
3.3.1	Treinamento supervisionado	37
3.3.2	Treinamento não supervisionado	38
3.3.3	Algoritmo de treinamento <i>backpropagation</i>	39
3.3.4	<i>Backpropagation</i> para as redes recorrentes	41
3.3.5	O algoritmo Rprop	47
3.3.5.1	Descrição do algoritmo	48
3.3.6	Algoritmo de parada antecipada	50
4	METODOLOGIA	53
4.1	O modelo ARIMA	53

4.2	Aspectos gerais	53
4.3	Dados	54
4.4	Seleção de entradas	55
4.5	Algoritmo de seleção de lags	56
4.6	Parametrização	58
4.6.1	Simulação	58
4.6.2	Redes neurais e treinamento	58
4.6.2.1	Discussão da quantidade de neurônios na camada oculta	59
5	RESULTADOS DAS SIMULAÇÕES	61
5.1	Submercado Sudeste/Centro-Oeste	62
5.2	Submercado Sul	66
5.3	Submercado Nordeste	70
5.4	Submercado Norte	74
6	CONCLUSÕES	79
6.1	Submercados	79
6.2	Desempenho das redes neurais versus modelo ARIMA	79
6.3	Desempenho da rede neural direta x rede neural recorrente	80
6.4	Desempenho da rede neural direta sem seleção de lags x com seleção de lags	80
6.5	Relevância do estudo e aplicação prática	80
6.6	Trabalhos futuros	81
	Referências	82

1 INTRODUÇÃO

1.1 Motivação

O setor elétrico brasileiro vem sofrendo grandes mudanças nas últimas décadas, afetando a maneira com que a energia elétrica é comercializada. Antes da reforma da década de 90, o Estado era o principal responsável pela geração e distribuição de energia em todo o país. Em uma área determinada, empresas estatais detinham exclusividade do provimento energia para os consumidores de varejo residencial, comercial e industrial, se responsabilizando pela geração, transmissão, comercialização e distribuição de energia elétrica.

Em meados da década de 90, houve uma grande reforma no setor elétrico brasileiro visando à otimização das atividades de geração, transmissão, distribuição, e comercialização de energia. Tal mudança foi motivada pela conjuntura político-econômica de dívida externa e pressão de grupos estrangeiros interessados no crescente mercado consumidor brasileiro. O estado brasileiro, incapaz de atender à crescente demanda de mercado e precisando pagar sua dívida, inicia uma política de privatizações para introduzir descentralização e competição no setor através da abertura de mercado.

Uma consequência importante das reformas no setor foi a mudança no panorama da comercialização de energia elétrica (CASTRO; LEITE, 2008). A maior flexibilidade da indústria permitiu o desenvolvimento de um mercado *spot* de energia elétrica, que, em suma, propicia maior variedade e mais ajustes nas transações. Houve um crescimento do mercado de energia de curto prazo, incentivado pelos preços atraentes apresentados até 2005.

Embora o mercado de energia de curto prazo tenha ganhado importância, vários fatores (físicos, econômicos etc) contribuem para que esse ambiente apresente significativa incerteza e volatilidade (MAYO, 2009). Entre os fatores que causam essa volatilidade pode-se destacar:

- Impossibilidade de armazenamento de energia para venda futura;
- Incerteza da demanda;
- Inelasticidade de preço da demanda;
- Dependência da hidrologia;
- A expansão da geração a fim de atender o crescimento da demanda requer longo prazo.

Dadas essas características, torna-se de suma importância desenvolver técnicas que possibilitem a previsão do preço de energia elétrica em curto prazo. Esse tipo de previsão permitirá a tomada de decisões importantes como ‘quando’ e ‘quanto’ comprar energia, que influenciarão significativamente a estratégia econômica dos agentes que atuam no mercado de energia elétrica.

Para valorar o preço da energia no mercado de curto prazo brasileiro é utilizado o preço de liquidação de diferenças (PLD), que é calculado considerando dados do operador nacional do sistema (ONS) e tem como finalidade a otimização da operação do Sistema Interligado Nacional (SIN). Existem poucos trabalhos publicados enfocando a previsão do PLD no mercado brasileiro de curto prazo.

Sendo a previsão do PLD o foco do trabalho, o problema tratado é um problema de previsão de série temporal, já que PLD é calculado semanalmente, sendo portanto uma série de dados ordenados no tempo.

Uma das principais técnicas de previsão de séries temporais é a rede neural artificial. Essa técnica tem como características a adaptabilidade e o aprendizado e consegue ter bons resultados na previsão envolvendo sistemas não lineares. Neste trabalho as redes neurais serão a principal ferramenta utilizada para a predição do preço da energia elétrica.

1.2 Objetivos

O objetivo deste trabalho é mostrar um modelo computacional para a predição da série temporal PLD um passo à frente e adicionalmente comparar o desempenho de algumas técnicas diferentes na abordagem desse problema. A previsão um passo à frente significa prever a próxima ocorrência da série temporal, no caso deste trabalho, o preço da energia na próxima semana, como será explicado mais detalhadamente no decorrer do texto.

Para a tarefa serão utilizadas as técnicas de redes neurais e correlação como ferramenta auxiliar. Serão utilizados os dados do operador nacional do sistema (ONS) como variáveis de entrada para os modelos. A previsão do PLD poderá auxiliar os participantes do mercado de energia de curto prazo brasileiro nas decisões sobre a compra ou venda de energia.

Os objetivos específicos são os apresentados abaixo:

- Comparar o desempenho de redes neurais diretas e recorrentes na previsão do PLD além de outra técnica como comparação externa, o modelo ARIMA;
- Analisar como a seleção das variáveis externas afeta o desempenho das redes neurais em cada submercado;

- Analisar de que maneira a seleção de lags nas variáveis externas pode influenciar o desempenho das redes neurais.

Obs: Lags são dados passados em relação ao dado atual. Por exemplo, analisando a presente semana, o lag de ordem 1 seria o dado da semana passada, lag de ordem dois seria representado pelo dado de duas semanas atrás e assim sucessivamente.

1.3 Revisão bibliográfica

Nos últimos anos, várias formas de abordar o problema da previsão do preço de energia foram apresentadas. Wan et al. (2014) utiliza um sistema híbrido de rede neural com treinamento por algoritmo ELM (*extreme learning machine*), além do método de máxima verossimilhança não para a previsão de um ponto como é de praxe, mas para o cálculo da probabilidade de um evento dado um certo intervalo de tempo. Wan obteve bons resultados com dados do mercado australiano de eletricidade.

Sharma e D.Srinivasan (2013) propuseram uma abordagem híbrida composta por redes neurais recorrentes e sistemas excitáveis acoplados, apresentando resultados satisfatórios em vários mercados. Chaâbane (2014) também propôs um sistema híbrido, utilizando redes neurais e o modelo ARFIMA (*Auto-Regressive Fractionally Integrated Moving Average*) obteve previsões bastante precisas no mercado europeu Nordpool.

Keynia (2012) utiliza um algoritmo de seleção de recurso para escolher as entradas para uma rede neural composta e compara o desempenho de sua técnica com outras recentes nos mercados da Califórnia e da Espanha obtendo bons resultados, porém com um sistema computacional complexo.

Mais especificamente sobre previsão de energia elétrica de curto prazo, podemos mencionar os seguintes trabalhos:

Hong e Wu (2012) propuseram um método em cascata no qual utilizam a *principal component analysis* (PCA) para selecionar as entradas da rede neural. Mostraram que na maioria dos testes nos mercados selecionados foi possível atingir melhor resultado comparado com a rede neural convencional.

Anbazhagan e Kumarappan (2014) também propuseram um método em cascata, sendo a entrada da rede neural selecionada pelo algoritmo DCT (*discrete cosine transforms*). Foi mostrado nesse trabalho que o desempenho atingido foi próximo ao estado da arte com menos gasto computacional. Os testes foram realizados em bases de dados dos mercados espanhol e de Nova Iorque.

Outra abordagem utilizada por Anbazhagan e Kumarappan (2013) foi a rede neural recorrente. Foi mostrado que ela apresenta desempenho ligeiramente inferior aos métodos híbridos, porém com menos complexidade computacional. Vale a observação de

que a complexidade da rede neural recorrente depende significativamente do tamanho do conjunto de treinamento e da quantidade de parâmetros escolhidos para compor a rede, ou seja, da arquitetura. Essa observação também é feita no trabalho em questão.

Chogumaira e Hiyama (2011) utilizaram um sistema híbrido em que os dados de entrada são clusterizados em intervalos de tempo para aplicação de lógica fuzzy, e depois submetidos a uma rede neural para efetuar previsões. Os resultados para o mercado australiano mostraram que essa técnica apresenta melhor desempenho em relação às que consideram a entrada da rede como uma simples série temporal contínua.

Gupta, Chawla e Chawla (2013) também utilizaram um sistema híbrido envolvendo redes neurais e lógica fuzzy. O sistema processa os dados em cascata utilizando as duas técnicas em blocos diferentes. Os testes no mercado indiano indicaram melhora no desempenho em comparação à utilização de redes neurais tradicionais.

Ghodsí e Zakerinia (2012) fazem uma comparação de desempenho entre os métodos ARIMA, redes neurais e lógica fuzzy na previsão do preço da eletricidade de curto prazo no mercado de Ontario. As conclusões indicam melhor performance para as duas últimas técnicas.

Embora o problema da previsão de energia elétrica de curto prazo seja bem explorado na literatura, praticamente não há estudos sobre o mercado brasileiro, sobre previsão do PLD. Apenas um trabalho foi encontrado:

Reston, Affonso e Limão (2014) utilizam um modelo híbrido em cascata no qual a técnica ARIMA é utilizada para pré-processamento das entradas da rede neural. Nesse artigo são utilizadas variáveis externas como entrada assim como no presente trabalho. Os resultados apresentados mostram a boa capacidade de previsão do modelo proposto para o mercado brasileiro.

Pode-se perceber que as redes neurais são de grande utilidade na abordagem do problema do preço de energia. Adicionalmente, as redes recorrentes são pouco utilizadas nesse tipo de problema, apenas nos últimos 2 anos houve algumas publicações nas quais esse tipo de rede foi utilizado. Finalmente, a importância do presente trabalho fica evidenciada também pela falta de estudos sobre previsão de energia de curto prazo no mercado brasileiro.

1.4 Estrutura do trabalho

O trabalho possui 6 capítulos organizados da seguinte maneira:

Capítulo 1: Apresenta o entendimento do problema, a motivação e os objetivos do trabalho, a revisão bibliográfica, além da organização e descrição dos capítulos.

Capítulo 2: Nesse capítulo é mostrado o panorama geral da comercialização de

energia elétrica no Brasil. O enfoque será no mercado de curto prazo com o detalhamento do cálculo do PLD, dos agentes envolvidos e das modalidades de contrato.

Capítulo 3: Contém o detalhamento geral sobre redes neurais, e as especificidades das redes utilizadas no trabalho, o uso de realimentação, o treinamento e os algoritmos de otimização.

Capítulo 4: Apresenta a metodologia utilizada para a solução do problema: arquitetura das redes neurais utilizadas, parâmetros de treinamento, simulação, escolha de variáveis externas, entre outros.

Capítulo 5: Mostra os resultados obtidos derivados das várias hipóteses apresentadas no capítulo 4, permitindo a avaliação de desempenho das técnicas empregadas na solução do problema.

Capítulo 6: Nesse capítulo são discutidos os resultados obtidos e as conclusões são apresentadas. Além disso, são discutidas as possibilidades de trabalhos futuros.

2 COMERCIALIZAÇÃO DE ENERGIA ELÉTRICA NO BRASIL

Este capítulo apresenta uma visão histórica do panorama de comercialização de energia elétrica no Brasil, além de detalhar o estágio atual do mercado. Será apresentado o detalhamento sobre o cálculo do PLD, cuja previsão é o principal objetivo deste trabalho.

2.1 Contexto Histórico

Como já foi mencionado, até os anos 90, o estado era o principal responsável pelo setor elétrico brasileiro. Vários fatores influenciaram a grande reforma em direção à abertura de mercado que se seguiria, entre os quais podemos destacar:

- Esgotamento da capacidade de investimento estatal devido à crise econômica mundial de 80 (altos juros, redução dos fluxos de crédito);
- Necessidade de melhora no desempenho do setor;
- Pressão de grupos estrangeiros interessados no mercado brasileiro;
- Endividamento do estado;
- Incapacidade de atender à demanda crescente.

O Brasil seguiu uma tendência mundial de reestruturação do setor elétrico. Países desenvolvidos haviam iniciado reformas em direção à descentralização do setor uma década antes, e assim como neles a desverticalização foi o meio para atingir o objetivo da eficiência. Houve várias privatizações, visando garantir os investimentos necessários para o aumento da oferta de energia, visto que o estado não era capaz de atender a demanda.

O propósito da desverticalização era criar competição no mercado pelo livre acesso à rede, de tal forma que os preços definidos estimulassem novos investimentos da geração à distribuição. Como outras consequências da desverticalização eram esperados o ganho de eficiência da indústria, a redução no preço da energia, e a atração do capital privado.

Segundo Souza (2008), a nova estrutura do setor elétrico brasileiro impôs a necessidade de controle regulatório no mercado. Para isso foram criados a ANEEL (agência nacional de energia elétrica) e o ONS. Foi criado também o ambiente para a realização das transações de compra e venda de energia elétrica, o MAE (mercado atacadista de energia elétrica).

A reforma implementada, no entanto, apresentou muitas falhas. O novo modelo proposto não foi capaz de atender às peculiaridades técnicas e institucionais do setor

elétrico, houve falhas de planejamento e o cronograma de reformas e de privatizações sofreu atrasos. Além disso, instalou-se uma tensão política, pois a necessidade das privatizações foi colocada em questão pela sociedade. Como consequência dos erros na reforma, ocorreu um grave racionamento de energia em 2001, complicando mais a situação, e então foram necessárias novas mudanças no setor elétrico brasileiro.

Foi criado então, o comitê de revitalização do setor elétrico brasileiro, cuja função era analisar o modelo em vigência e propor medidas de aperfeiçoamento e correção. Estudos foram realizados e as medidas que se seguiriam constituíram o novo modelo de comercialização de energia elétrica no Brasil.

2.2 Novo marco regulatório

Em 2003 iniciou-se a reestruturação do setor elétrico brasileiro, impulsionada pela nova estrutura política resultante do processo eleitoral. As leis 10.847 e 10.848 de 2004 foram as bases institucionais e legais para o novo modelo.

As novas medidas visavam garantir o suprimento de energia elétrica a toda a população e a modicidade tarifária. Elas impuseram a obrigatoriedade de procedimentos públicos transparentes para a comercialização de energia pelas geradoras estatais. As distribuidoras não poderiam mais negociar energia livremente sendo obrigadas a participar de leilões, observado o critério de menor tarifa.

O novo modelo não permitia a verticalização das empresas de distribuição, não sendo possível uma distribuidora ter propriedade de ativos de geração. Essa característica do modelo reduz os potenciais subsídios cruzados e contribui para a modicidade tarifária, facilitando o acesso à energia por parte da população em geral.

Outras medidas foram tomadas visando a qualidade e a segurança do serviço de fornecimento de energia entre as quais vale a pena destacar:

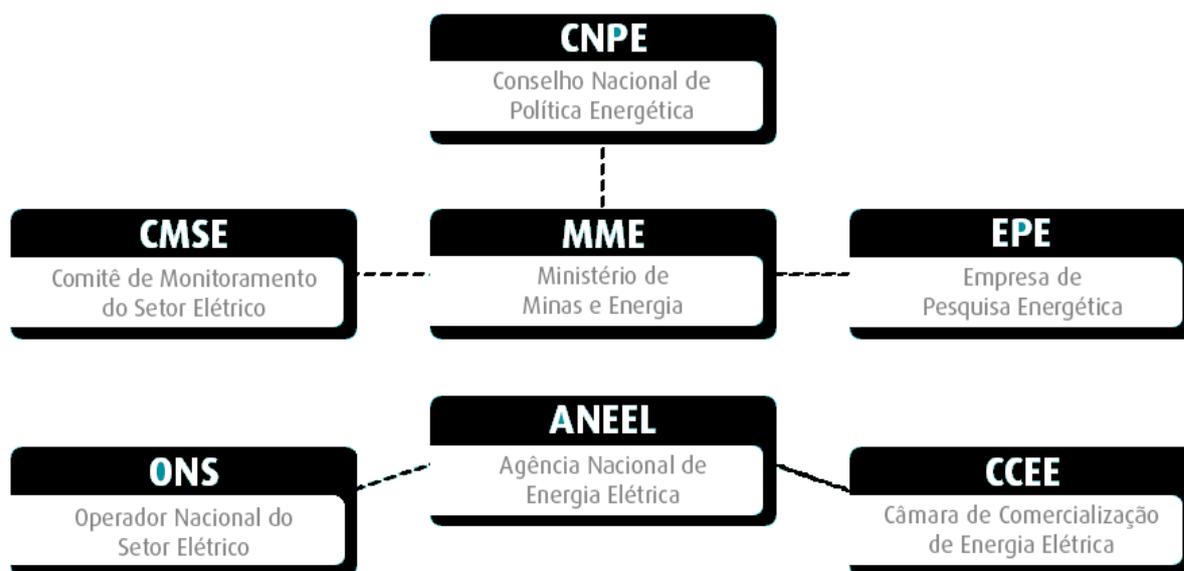
- Cobertura contratual de 100% do consumo de energia tanto pelas distribuidoras quanto pelos consumidores livres;
- Contratação de usinas hidrelétricas e termelétricas em proporções que assegurem equilíbrio entre garantia e custo de suprimento;
- Exigência de licenças ambientais para se permitir a participação em um novo empreendimento no processo de licitação.

Juntamente com as novas medidas, houve a necessidade de criação de novos órgãos para regular o mercado de energia elétrica. A Empresa de Pesquisa Energética (EPE) foi criada com o objetivo de planejar o setor elétrico em longo prazo, o comitê de

monitoramento do setor elétrico (CMSE), criado para continuar às atividades do MAE (Mercado Atacadista de Energia) relativas à comercialização de energia elétrica no sistema interligado.

A Figura 1 abaixo mostra as instituições do setor elétrico brasileiro:

Figura 1 – Instituições do setor elétrico brasileiro.



Site da CCEE: www.ccee.org.br

A Câmara de comercialização de Energia Elétrica foi criada pela lei nº 10.848, de 15 de março de 2004 e regulamentada pelo decreto Nº 5.177 de 12 de Agosto de 2004 sucedendo o Mercado Atacadista de Energia e herdando suas estruturas operacionais e organizacionais. Suas atribuições principais são: O cálculo do Preço de Liquidação de Diferenças (PLD), utilizado para valorar as transações no mercado de curto prazo; a contabilização dos montantes de energia elétrica comercializados; a realização de leilões de compra e venda de energia elétrica no Ambiente de Contratação Regulada (ACR) e a liquidação dos valores decorrentes das operações de compra e venda de energia realizadas no mercado de curto prazo.

Diversas regras e procedimentos regem a operação comercial da CCEE e os processos de contabilização e liquidação financeira. As regras são um conjunto de equações matemáticas e fundamentos conceituais que complementam a convenção de comercialização. Já os procedimentos de comercialização são normas que definem condições, requisitos, eventos e termos relacionados à comercialização de energia elétrica. Tanto as regras quanto os procedimentos estão sujeitos à aprovação da ANEEL. Todos os agentes com participação prevista na convenção de comercialização de energia elétrica são associados da CCEE.

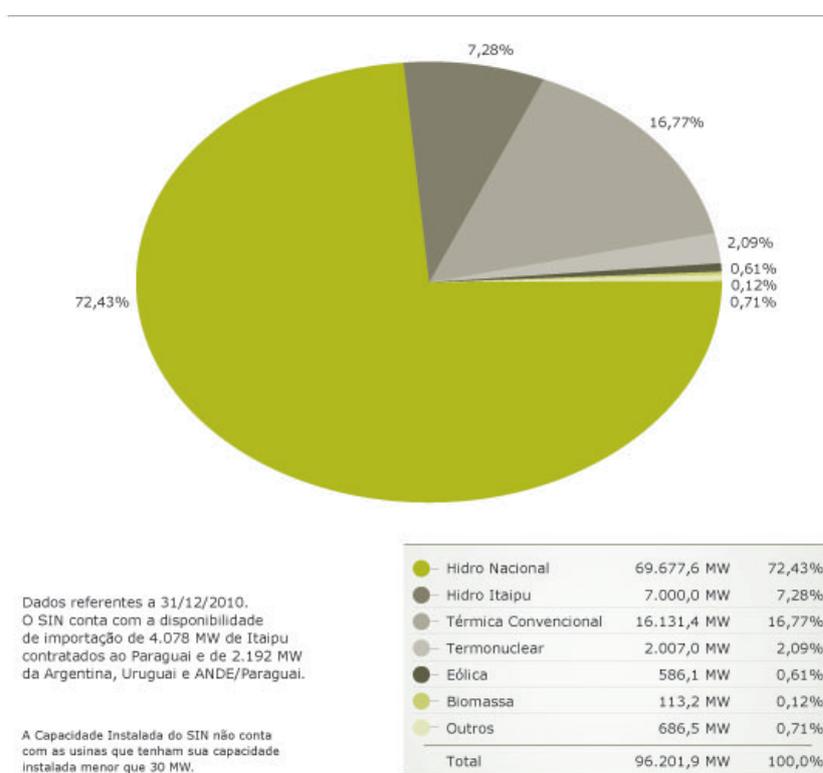
2.3 O sistema interligado nacional

Segundo o site da ONS, o sistema de produção e transmissão de energia elétrica no Brasil é um sistema hidrotérmico de grande porte, com forte predominância de usinas hidrelétricas e com vários proprietários. O sistema brasileiro possui tamanho e características que permitem considerá-lo único em âmbito mundial. O SIN é formado por empresas das regiões Sul, Sudeste, Centro-Oeste, Nordeste e parte da região Norte. A grande maioria da energia elétrica utilizada no país provém do SIN, com apenas 1,7% sendo fornecido por pequenos sistemas de fora, localizados principalmente na região amazônica.

Ao final de 2014, o SIN tinha uma capacidade instalada de geração de mais de 133.000 MW e mais de 120.000 km de linhas de transmissão em tensão igual ou superior a 230 kV. O Brasil é considerado um dos países emergentes do Grupo BRICS (Brasil, Rússia, Índia, China e África do sul), cuja economia e demanda por energia elétrica têm crescido significativamente a cada ano.

A Figura 2 mostra a estrutura da capacidade instalada no SIN por fonte de energia, dados de 31/12/2010 segundo o site do ONS.

Figura 2 – Estrutura da capacidade instalada do SIN por fonte de energia.

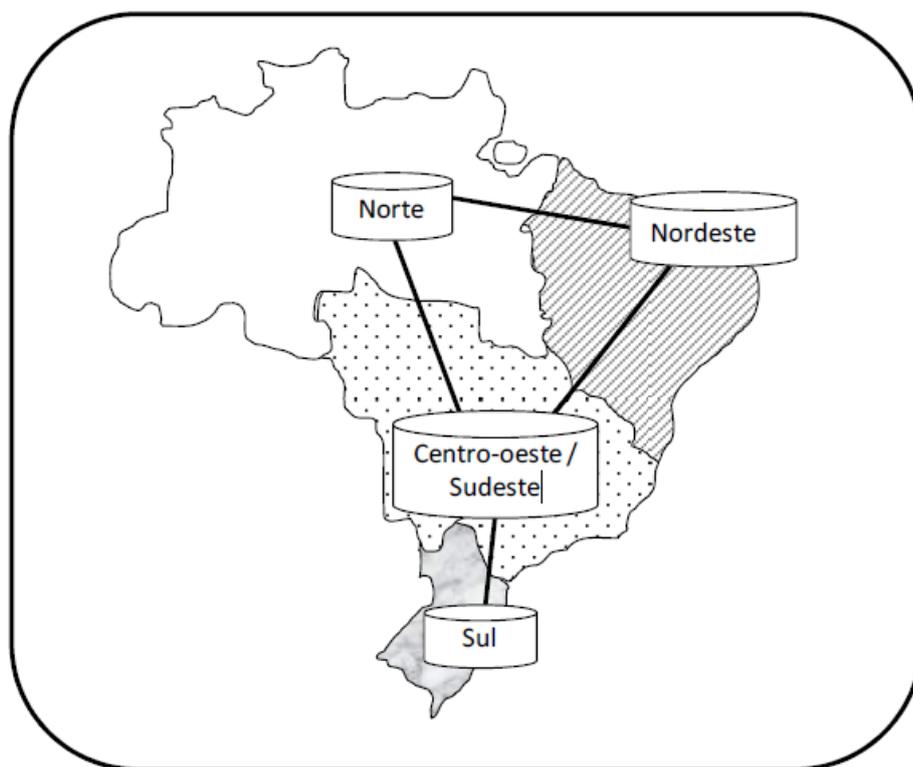


Site do ONS: www.ons.org.br

As linhas de transmissão são responsáveis por transportar energia entre as regiões do país. A diversidade hidrológica existente entre as regiões permite a redução dos custos

operativos do sistema, entretanto, devido à restrições de transmissão entre as regiões, o mercado de eletricidade brasileiro foi dividido em 4 submercados: Norte, Nordeste, Sudeste/Centro-Oeste e Sul. A figura 3 a seguir ilustra os submercados brasileiros.

Figura 3 – Submercados de energia no brasil.



www.ccee.org.br

2.4 Ambientes de contratação

O novo modelo criou dois ambientes com lógica e estrutura distintas de comercialização de energia elétrica. O primeiro, centrado em consumidores cativos, que são consumidores residenciais, de serviço ou indústria com nível baixo de consum. Este é o Ambiente de Contratação Regulada (ACR). O segundo é voltado para grandes volumes de consumo e muitas vezes em curto prazo, onde prevalecem empresas de grande porte denominadas consumidores livres. Este mercado é denominado ambiente de contratação livre (ACL).

O Ambiente de Contratação Regulada é regido por tarifas e condições de fornecimento ditadas pela ANEEL. A comercialização é feita através de contratos resultantes de leilões, não sendo possíveis alterações bilaterais pelos agentes (MAYO, 2009). Os contratos são celebrados entre agentes vendedores (comercializadores, geradores, produtores independentes e autoprodutores) e compradores (distribuidores).

No Ambiente de contratação Livre, os agentes podem negociar contratos bilaterais

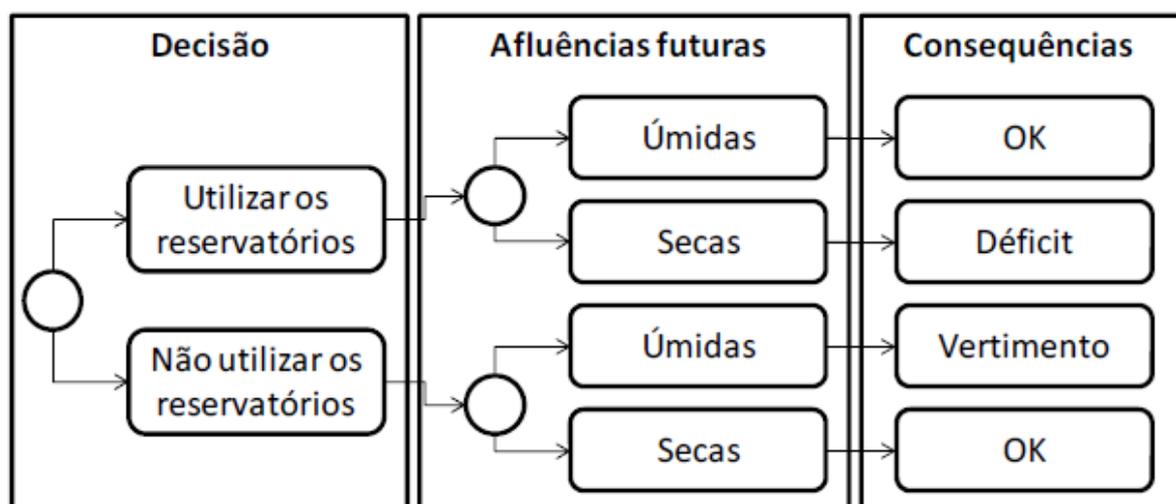
livremente, não existindo restrições de volume, preços ou prazos. Entretanto, os contratos devem ser registrados na CCEE para a realização da liquidação financeira das diferenças entre os montantes contratados e o efetivamente consumido. Participam do ACL os Agentes de geração, comercializadores, importadores e exportadores de energia e consumidores livres.

2.5 Modelo de despacho centralizado (*tight pool*)

Para entendimento do modelo *tight pool*, a outra opção de modelo, o *loose pool* será explicada resumidamente: O modelo *loose pool* é utilizado quando a predominância da geração de energia elétrica é de termelétricas, pois os preços de geração são mais previsíveis, já que o insumo que irá gerar a energia é um combustível (óleo, carvão, gás natural etc) que tem preço conhecido. Nesse modelo, geradores e consumidores propõem volumes e preços e o próprio mercado regula o preço da energia pela lei de oferta e procura.

Já no modelo *tight pool*, o custo da geração de energia é muito mais complexo. O objetivo do modelo é evitar ao máximo a utilização da geração térmica, que é mais cara. Para isso é necessário uma utilização otimizada dos recursos hidrológicos, o que envolve um processo de decisão complexo. Existem riscos e consequências associados às decisões tomadas em relação à utilização dos recursos hidrológicos, já que a capacidade dos reservatórios é limitada e a afluência futura é incerta (SILVA, 2001). A Figura 4 mostra o sistema de decisão mencionado:

Figura 4 – Processo de decisão em um sistema hidrotérmico.



Silva 2001

Dessa forma, se a decisão de utilizar os reservatórios for tomada, as consequências podem ser: Positivas se as afluências futuras forem altas (período chuvoso) representando

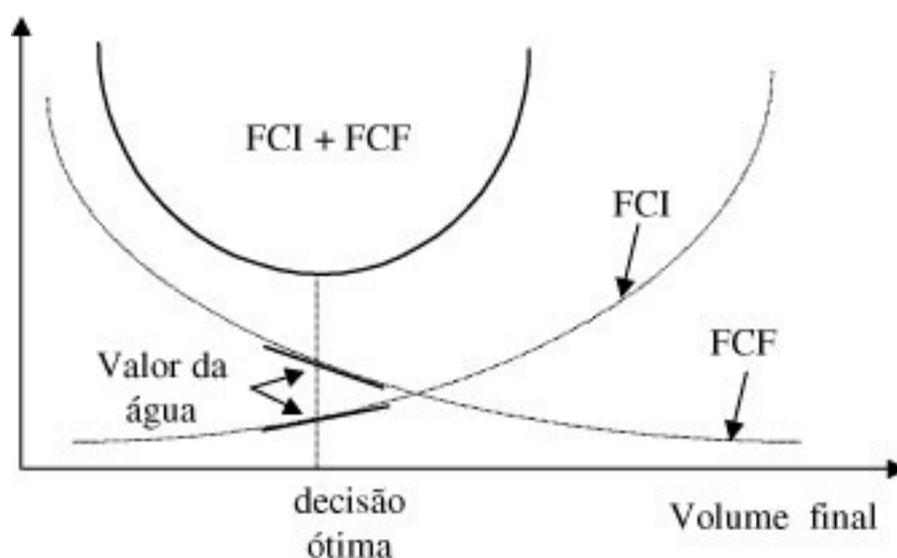
uma operação econômica, ou negativas, se as afluências futuras forem baixas (período seco) representando uma operação com déficit. Caso a decisão tomada seja a de não utilizar os reservatórios, as consequências podem ser: Positivas se as afluências forem baixas, representando uma operação econômica, ou negativas em caso de afluências futuras altas, nesse caso há vertimento, que não é tão danoso quanto um déficit, porém há desperdício de energia

O ONS é responsável por comparar o benefício do uso imediato dos recursos hídricos com o benefício futuro do armazenamento dos mesmos. Essa tarefa é representada matematicamente pelas funções de custo imediato e de custo futuro (FCI e FCF).

A função de custo imediato representa os custos da geração térmica no momento presente. O custo imediato aumenta à medida que o volume de energia hidráulica disponível diminui, assim, quanto menos a fonte hidroelétrica for utilizada, maior a necessidade do uso da geração térmica. Já a função de custo futuro diminui à medida que aumenta o volume armazenado, pois dessa forma diminui a necessidade futura da utilização da geração térmica, pois haverá recursos hidrológicos disponíveis.

O ponto de decisão ideal é aquele que minimiza a soma dos custos imediato e futuro. A Figura 5 ilustra o ponto de decisão ótimo, que é onde as derivadas das duas funções se igualam em módulo.

Figura 5 – Custos de geração imediato e futuro e decisão ótima



Silva 2001

O sistema busca utilizar os recursos de maneira otimizada para diminuir os custos da geração, sempre objetivando a modicidade tarifária.

2.6 Mercado de curto prazo

Devido às incertezas envolvidas no processo de geração e comercialização de energia elétrica, podem existir diferenças entre o montante de energia produzido e o consumido. Essa diferença é comercializada no mercado de energia de curto prazo. Diferenças positivas ou negativas são liquidadas no mercado de curto prazo e valoradas ao PLD, que é calculado semanalmente para cada submercado e cada patamar de carga.

O mercado de curto prazo pode ser definido como o segmento da CCEE onde são contabilizadas as diferenças entre os montantes de energia elétrica contratados pelos agentes e os montantes de geração efetivamente verificados. No mercado de curto prazo não existem contratos, e sim contratação multilateral, conforme as regras de comercialização da CCEE.

2.7 Preço de liquidação de diferenças (PLD)

O PLD é utilizado para valorar a energia comercializada no mercado de curto prazo e usa dados considerados pelo ONS para a otimização da operação do SIN. O cálculo do PLD tem o objetivo de encontrar a solução ótima de equilíbrio entre a utilização imediata da água e o benefício de seu armazenamento futuro, visando economizar a geração de energia das usinas termelétricas.

O PLD é um valor determinado semanalmente para cada patamar de carga (leve, médio e pesado) e para cada submercado (Norte, Nordeste, Sudeste/Centro-Oeste e Sul). Seu cálculo é baseado no Custo Marginal de Operação (CMO) e é limitado por um valor máximo e um mínimo vigentes em cada período de apuração e determinados pela ANEEL.

O CMO é um resultado intermediário, uma etapa do processo de cálculo do PLD. Ele representa a variação do custo operativo necessário para atender um MWh adicional de demanda, utilizando os recursos existentes. O PLD é determinado com base no CMO para a semana do horizonte de planejamento, considerando-se os limites definidos pela ANEEL.

O processamento realizado pela CCEE suprime as restrições internas de transmissão internas, de forma que a energia é considerada igualmente disponível em qualquer ponto e conseqüentemente seu preço uniforme em cada submercado. No cálculo do PLD são apenas consideradas as restrições de transmissão entre os submercados.

O cálculo do PLD é realizado por modelos matemáticos baseados nas seguintes variáveis:

- Demanda futura;
- Previsão hidrológica;

- Disponibilidade dos sistemas de geração e transmissão;
- Custo de déficit;
- Entrada em operação de novas plantas.

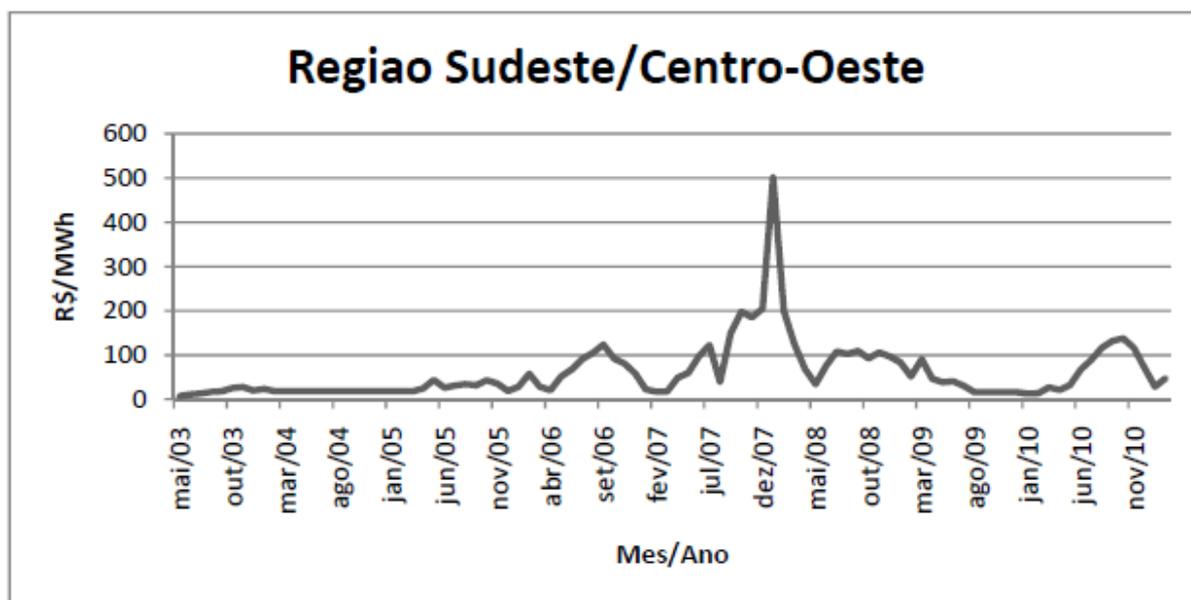
O resultado é o despacho ótimo no horizonte de estudo para ambos os tipos de geração (hidráulica e térmica) e para cada submercado (Norte, Nordeste, Sudeste/Centro-Oeste e Sul).

O modelo matemáticos utilizado para o cálculo do PLD usa os programas NEWWAVE, NEWDESP e DECOMP. Maiores detalhes podem ser encontrados no site da CCEE www.ccee.org.br. A seguir é apresentada uma breve descrição de cada um e sua função:

- NEWWAVE: Modelo de otimização focado no médio prazo (horizonte de cinco anos), discretizado mensalmente. Seu objetivo é determinar a estratégia para uso da geração hidráulica e térmica em cada estágio, que minimiza o custo de operação para todo o período de planejamento. Os principais resultados do modelo são as funções de custo futuro, que traduzem para as outras etapas (de curto prazo) o impacto da utilização da água armazenada nos reservatórios.
- NEWDESP: Programa componente do sistema NEWWAVE que serve para consulta às funções de custo futuro geradas pelo módulo de otimização (o próprio NEWWAVE). O programa pode executar em dois modos, a consulta e o despacho. No modo consulta, um relatório com os valores de água para aquele mês é gerado com base nas informações de energia armazenada no final do mês passado e energias afluentes realizadas para cada um dos subsistemas estudados no NEWWAVE. Já no modo despacho, o modelo determina o despacho ótimo para o período em estudo, definindo a geração hidráulica equivalente e o despacho das usinas térmicas para cada subsistema. O modo despacho fornece diretamente os preços da CCEE por patamar de carga para cada submercado.
- DECOMP: Modelo de otimização de curto prazo (horizonte de 12 meses), que representa o primeiro mês em base semanal, as vazões previstas, a aleatoriedade das vazões do restante do período através de uma árvore de possibilidades (cenários de vazões) e o parque gerador individualizado (usinas hidráulicas e térmicas por subsistema). O seu objetivo é determinar o despacho ótimo de geração para as usinas hidráulicas e térmicas ao longo do período de planejamento dadas as informações disponíveis (carga, vazões, disponibilidade, limites de transmissão entre subsistemas, função de custo futuro do NEWWAVE). Os principais resultados desse modelo são os despachos de geração por usina hidráulica e térmica de cada submercado, e os custos marginais de operação para cada estágio e cada patamar de carga.

O sistema de geração de energia brasileiro sendo baseado principalmente em hidrelétricas com grandes reservatórios é projetado para assegurar o fornecimento de energia mesmo em condições hidrológicas adversas, que não são frequentes. Dessa forma durante a maior parte do tempo os reservatórios estão cheios, resultando em PLD's baixos. Entretanto, quando um período seco ocorre, o PLD pode atingir valores bastante elevados. A figura 6 mostra o comportamento do PLD para o submercado Sudeste/Centro-Oeste:

Figura 6 – PLD no mercado sudeste/centro-oeste



Site da CCEE: www.ccee.org.br

A seguir no capítulo 3 serão detalhadas as redes neurais artificiais e as especificidades implementadas neste trabalho como algoritmos de treinamento e arquitetura.

3 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais (RNA) são uma técnica de computação inspiradas no funcionamento do cérebro humano. O estudo das redes neurais representa um outro paradigma na modelagem de problemas, e se constitui numa boa alternativa na solução computacional de vários deles. As redes neurais apresentam como principais características a adaptação e o aprendizado, pois conseguem armazenar conhecimento baseado em experiências e torná-lo disponível para uso posterior. As RNA são formadas por conjuntos de unidades simples de processamento chamadas de *neurônios artificiais*, organizadas em uma ou mais camadas, e interligadas por conexões chamadas de *pesos sinápticos*. Esses pesos são ajustáveis e são modificados no processo de aprendizagem da rede, eles são responsáveis por armazenar o ‘conhecimento’ da rede.

A inspiração no cérebro humano surgiu após estudos biológicos comprovarem que o cérebro executa sua computação de uma maneira completamente diferente da de um computador digital convencional. O cérebro realiza tarefas através de uma grande e complexa rede de unidades de processamento chamadas de neurônios. Hoje sabe-se que a computação realizada no nosso cérebro é não-linear e paralela. As conexões entre neurônios são mutáveis, novas conexões são formadas de acordo com a aprendizagem, e outras podem morrer sem que o processamento global seja prejudicado perceptivelmente (HAYKIN, 1999).

O cérebro humano, após certo grau de maturidade, é capaz de resolver problemas com os quais nunca teve contato, e mais que isso, problemas para os quais não foi diretamente preparado. Um exemplo dessa capacidade seria o reconhecimento de uma pessoa com a qual deixamos de conviver há muito tempo, ou identificar formas de escrita (letra) de pessoas diferentes, tarefas que são computacionalmente muito difíceis. Sabe-se que a capacidade de identificação do cérebro humano é adquirida com o tempo, num processo chamado de aprendizado por experiência. Os modelos de RNA tentam reproduzir essa característica biológica.

As RNA diferem do processamento convencional, pois não obedecem à estrutura linear e seqüencial da computação algorítmica (BRAGA; CARVALHO; LUDERMIR, 2000). Podem-se citar as características biológicas que as redes neurais artificiais também possuem:

- É composta de várias unidades de processamento;
- Processamento é paralelo (cada unidade opera independentemente);
- Alta interconexão entre as unidades de processamento;

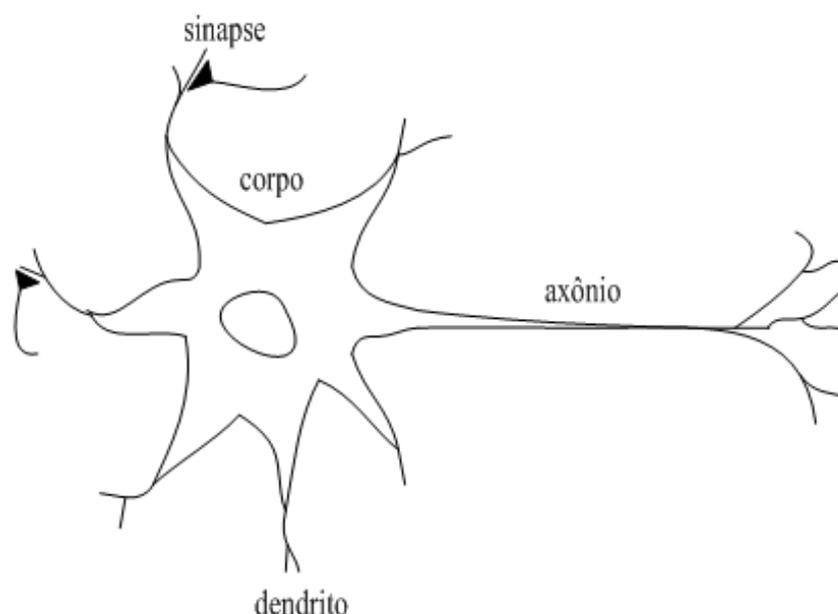
- Existência de “aprendizado”;
- Uso do conhecimento adquirido através de exemplos.

Assim como a rede neural biológica, a artificial também passa por um processo de aprendizagem. Uma forma de aprendizagem das RNA consiste na análise de exemplos, em que são fornecidos para a rede padrões de entrada com as respectivas saídas desejadas. É executado então um algoritmo de *treinamento* que ajusta os pesos sinápticos com o objetivo de fazer a rede produzir saídas as mais próximas possíveis das saídas desejadas, na esperança de que a rede também produza saídas aceitáveis para novas entradas não utilizadas durante a fase de treinamento.

3.1 Neurônio biológico x neurônio artificial

Na Figura 7 pode-se ver a estrutura do neurônio biológico. Os dentritos são responsáveis por coletar os estímulos nervosos que servirão de entrada para o neurônio. O corpo celular é responsável pela computação dos dados coletados e por gerar a saída, que servirá de estímulo para outros neurônios. O axônio é responsável por transmitir a saída gerada pelo neurônio, ele possui diversas terminações que podem se conectar a outros neurônios formando assim a rede biológica humana.

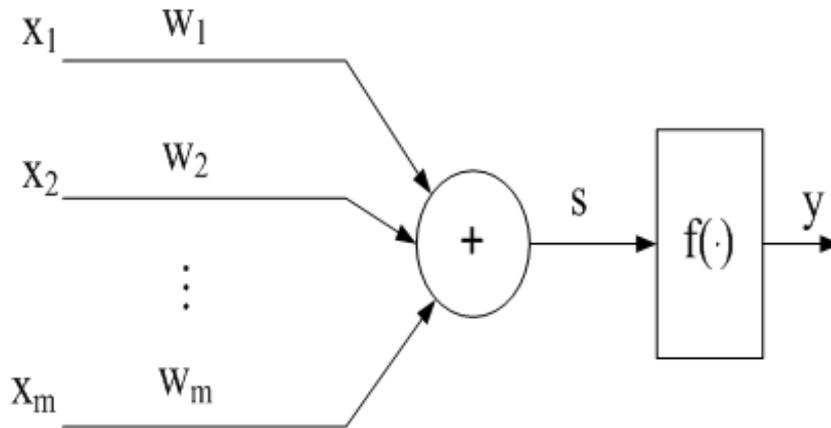
Figura 7 – Estrutura do neurônio biológico



Obviamente a descrição aqui feita está num nível bastante superficial dado o objetivo deste trabalho. A transmissão de informação entre neurônios biológicos é bem mais complexa, um neurônio pode receber estímulo de até 10.000 outros neurônios, e pode fornecer estímulo para outros milhares (VEELENTRUF, 1995).

O modelo de neurônio artificial é inspirado no neurônio biológico. Pode-se ver sua estrutura apresentada na Figura 8:

Figura 8 – Modelo de neurônio artificial



A descrição matemática do neurônio apresentado pela Figura 2 é feita pela Equação 3.1 apresentada abaixo, em que x são as entradas do neurônio e w os pesos sinápticos correspondentes:

$$y = f\left(\sum_{n=1}^m x_n \cdot w_n\right) \quad (3.1)$$

A computação no neurônio artificial ocorre da seguinte maneira: as entradas são multiplicadas pelos *pesos sinápticos* e posteriormente somadas na entrada do neurônio, formando uma soma s das entradas ponderadas pelos pesos sinápticos. Essa soma passa por uma *função de ativação* e o resultado é então a saída do neurônio. Como na rede neural biológica, a saída de um neurônio pode servir de entrada para outros neurônios.

3.2 Tipos de redes neurais

A classificação apresentada aqui diz respeito ao uso de realimentação na arquitetura da rede neural. Quanto a essa utilização as redes podem ser diretas (estáticas) ou recorrentes (dinâmicas).

3.2.1 Redes neurais diretas

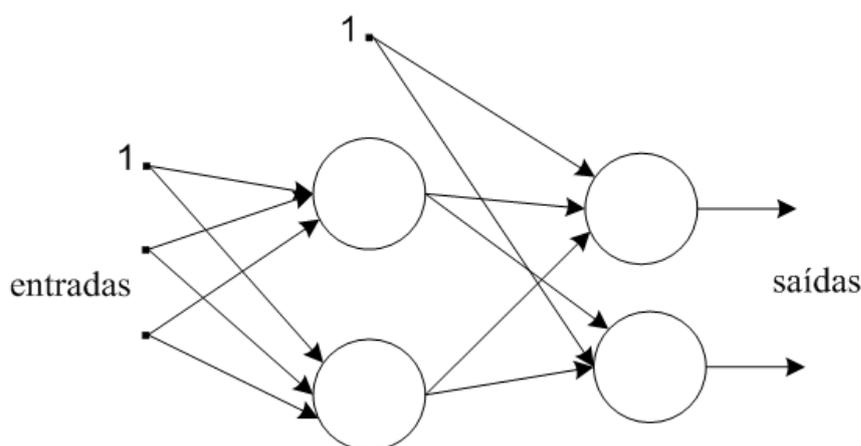
As redes estáticas ou diretas ou redes multi layer perceptron (MLP) são aquelas que não possuem conexões de realimentação, ou seja, a saída de um neurônio só pode

alimentar outros em camadas posteriores na rede. O processamento ocorre de maneira direta, um valor gerado por um neurônio não pode ser usado em camadas anteriores àquela na qual o valor foi gerado, também não pode ser usado na mesma camada.

Uma consequência importante dessa definição, é que cada conjunto de valores de entrada influenciará apenas na saída gerada naquele instante de processamento, ou seja, valores de saída futuros são gerados por valores de entrada futuros apenas. Outra consequência da definição, é que um conjunto de entradas sempre produzirá a mesma saída, independentemente do instante de tempo em que esse conjunto foi processado, ou seja, uma rede direta não possui estados internos logo não é capaz de oferecer representações dinâmicas internas do processo que se quer modelar.

Um exemplo de RNA estática pode ser visto na Figura 9 abaixo:

Figura 9 – Exemplo de RNA direta



Outra maneira de analisar se dá por meio da analogia com funções matemáticas. Um conjunto de variáveis passa pela função e gera um conjunto de saída. Se conjuntos de entradas iguais são aplicados à função, ela produzirá o mesmo resultado não importando o tempo em que a operação for feita. Dessa forma pode-se dizer que a rede neural estática é um sistema sem memória, capaz de fazer apenas mapeamentos estáticos.

Embora a rede neural direta seja capaz de realizar apenas mapeamentos estáticos, seu processamento, treinamento e parametrização é bem mais simples em relação às redes recorrentes. Dessa forma, dependendo da característica do problema a ser tratado, a rede direta pode ser a melhor opção a ser utilizada.

3.2.2 Redes neurais recorrentes

As redes neurais recorrentes ou dinâmicas possuem conexões de realimentação. A saída gerada por uma unidade pode alimentar outras unidades da mesma camada ou de

camadas anteriores ao longo do processamento. O uso de realimentação nessas redes gera fortes implicações.

Quando um conjunto de entradas é processado, as conexões de realimentação continuam ativando as unidades de processamento em instantes de tempo posteriores. Assim, mesmo quando a entrada cessa a rede continua produzindo saídas. A rede agora apresenta um *estado interno*, a saída não é determinada apenas pelo conjunto de entradas, mas também pelo estado da rede (entradas utilizadas no passado). O mesmo conjunto de entradas pode gerar resultados diferentes dependendo do estado da rede.

Essas características das redes recorrentes demonstram um forte potencial desse tipo de rede para identificação de sistemas dinâmicos, já que a rede agora possui uma memória, que permite o processamento de informação de contexto temporal (LUKOŠEVIČIUS; JAEGER, 2009).

3.2.2.1 Arquiteturas de redes neurais recorrentes

Como já foi mencionado, as RNAs recorrentes apresentam característica dinâmica devido às suas conexões de realimentação. Porém essas realimentações podem estar presentes de diversas formas na estrutura da rede, que se diferenciam na maneira de representar o estado interno da rede.

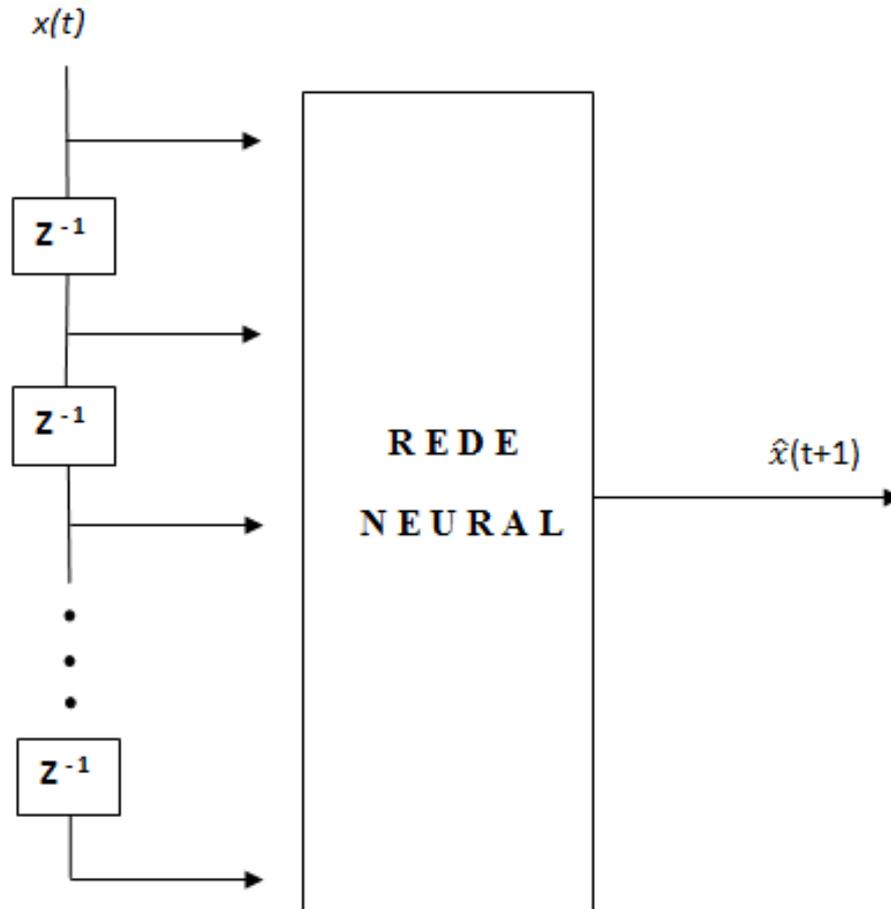
Williams (1990) classifica as representações temporais desenvolvidas pelas redes neurais de acordo com a sua complexidade e flexibilidade. A representação temporal pode estar presente na rede de várias formas. Williams descreve três abordagens para a representação temporal nas redes neurais: A abordagem *conservadora*, a abordagem *liberal*, e a abordagem *radical*. Descreve-se nas subseções seguintes cada uma com mais detalhes:

3.2.2.1.1 Abordagem conservadora

Uma maneira mais “direta” de se inserir dinâmica na rede neural é usando uma linha de atraso na entrada. A rede neural será alimentada pelo valor da entrada atual, e também valores dessa entrada atrasados no tempo. O uso de um número adequado de entradas atrasadas permite a captura de informações de estado. Essa abordagem, na qual a entrada e seus valores passados alimentam a rede, é chamada de *abordagem conservadora*.

A abordagem conservadora é bastante comum quando o sinal a ser processado é variante no tempo. Apesar de existir informação de dinâmica, essa informação é externa, logo, essa abordagem não requer o uso obrigatório de uma rede recorrente para a sua implementação, o que pode facilitar o treinamento já que envolve nada mais que treinar uma rede neural estática. Pode-se ver o modelo de abordagem conservadora na Figura 10.

Figura 10 – Abordagem conservadora

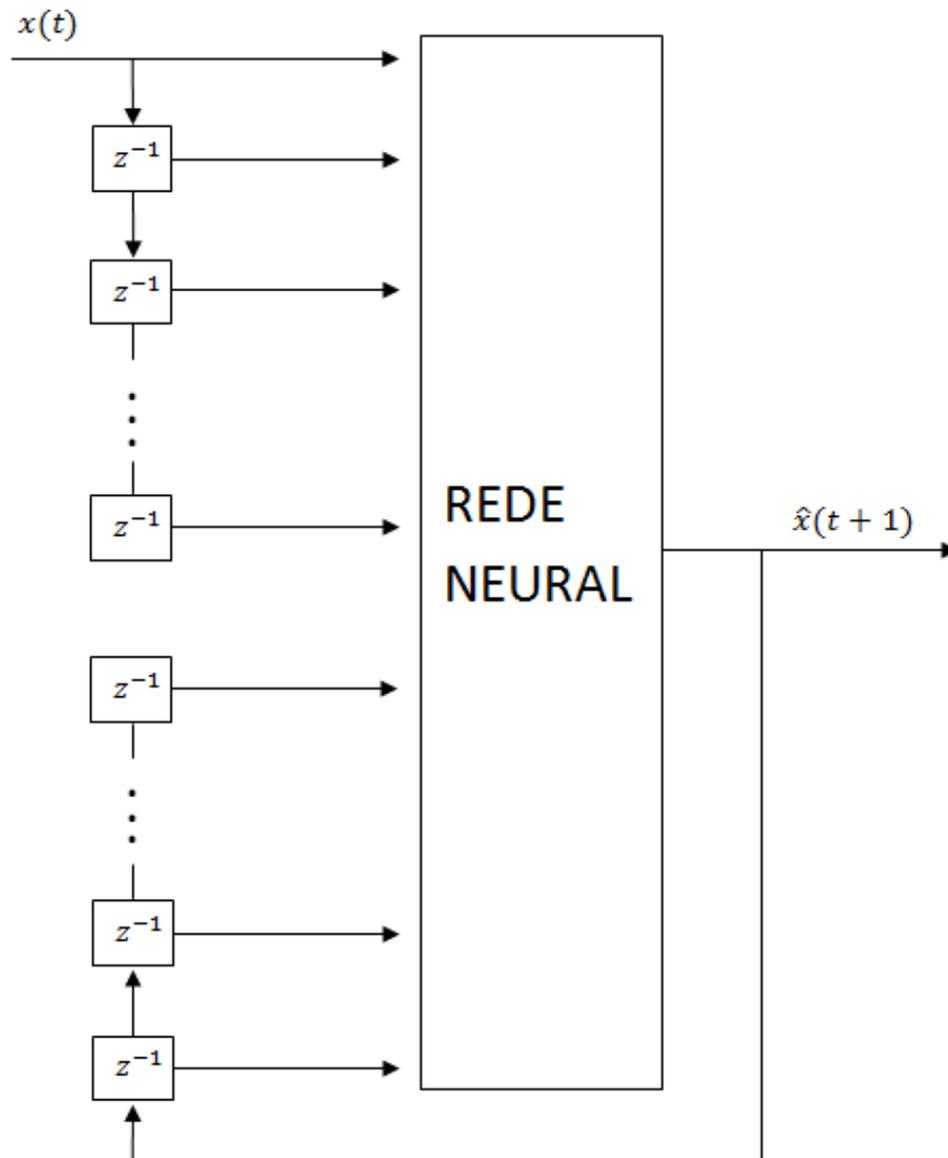


3.2.2.1.2 Abordagem liberal

A abordagem liberal é aquela na qual são usadas linhas de atraso tanto na entrada quanto na saída da rede, que dessa forma, é alimentada pelas versões atrasadas da entrada e da própria saída gerada. Em alguns sistemas, a informação de estado precisa dos valores de entrada e saídas geradas para ser representado.

Essa arquitetura também é chamada na literatura de NARX (Nonlinear Autoregressive model process with eXogenous input) pois utiliza valores passados da própria saída da rede como uma nova entrada para a iteração seguinte. Pode-se ver o modelo de abordagem liberal na Figura 11.

Figura 11 – Abordagem liberal

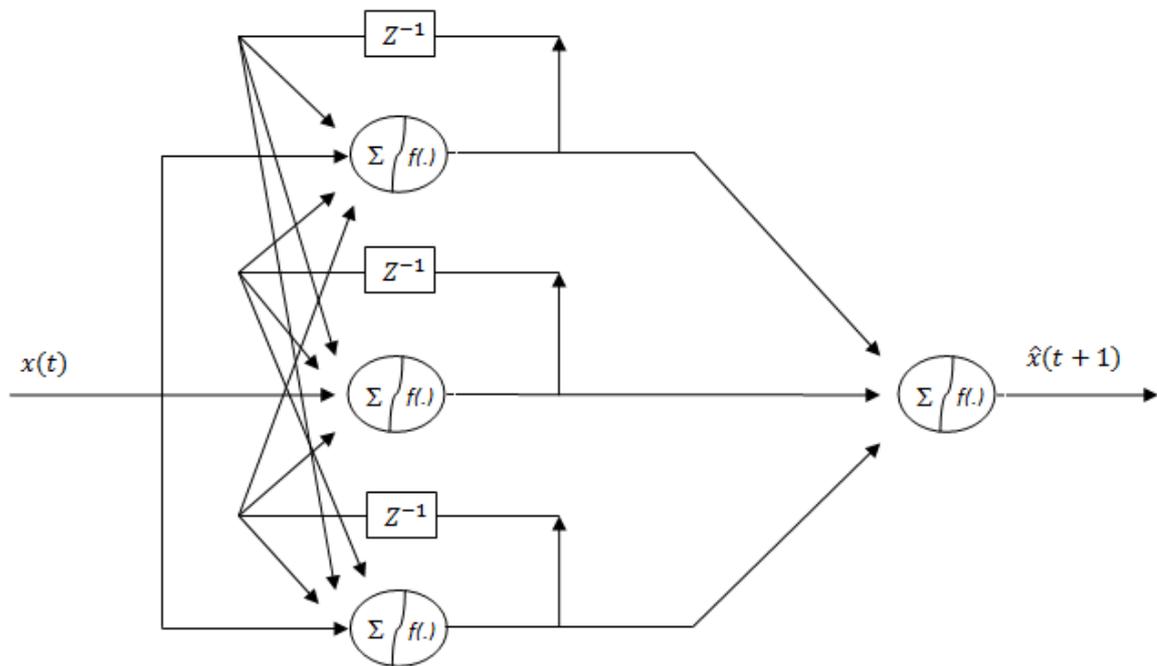


3.2.2.1.3 Abordagem radical

Na abordagem radical a dinâmica está presente internamente à rede. As conexões de realimentação estão nos neurônios *ocultos* da rede, ou seja, que não são de entrada e nem de saída, são unidades de processamento intermediário. Essa abordagem é necessária quando o sistema ou planta a ser modelado possui *forte estado interno oculto*, ou seja, seu estado não pode ser representado apenas por uma função de uma quantidade finita de valores passados de sua entrada e de sua saída (WILLIAMS, 1990).

Esse tipo de arquitetura corresponde a rede de Elman utilizada em Pham e Liu (1996) para a modelagem de sistemas dinâmicos. A rede de Elman também possui conexões de realimentação internas, presentes nos neurônios ocultos da rede. Pode-se ver o modelo de abordagem radical na Figura 12.

Figura 12 – Abordagem radical



As linhas de atraso na entrada ou na saída usadas nas abordagens conservadora e liberal representam o estado de uma forma pré-determinada, enquanto que na abordagem radical a rede forma as suas próprias representações de estado. De certa forma pode-se dizer que a abordagem radical constitui-se em um modelo “mais” adaptativo ou mais flexível que os oferecidos pelas abordagens conservadora e liberal.

3.3 Treinamento de redes neurais

Como já foi mencionado, as RNA possuem a capacidade de “aprender”. O processo de aprendizagem, de modo geral, ocorre com a apresentação de exemplos para a rede neural. Com base nos exemplos fornecidos a rede ajusta seus pesos sinápticos com o objetivo de aproximar a saída gerada da saída desejada em um processo denominado treinamento. Existem duas grandes classes principais de treinamento para RNA, o treinamento supervisionado (que é empregado neste trabalho) e o não supervisionado, apresentadas a seguir:

3.3.1 Treinamento supervisionado

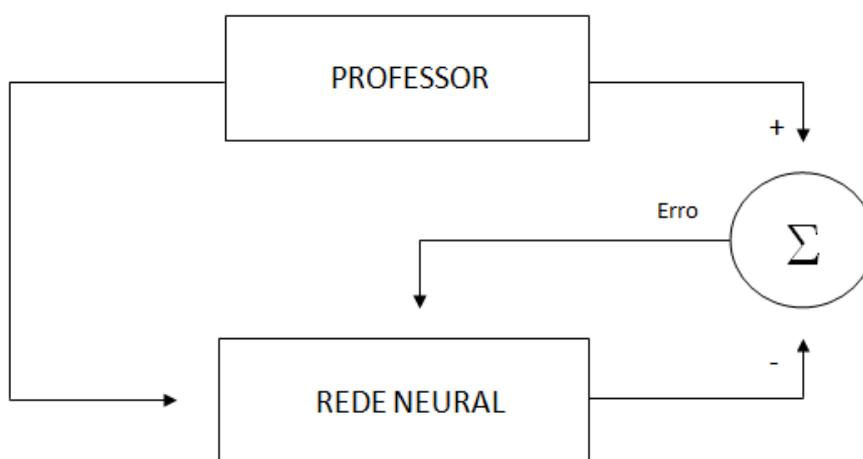
No treinamento supervisionado, é oferecido para a rede um conjunto de treinamento que contém entradas e as respectivas saídas desejadas. Pode-se conceber que estes padrões são oferecidos por um agente externo denominado “professor”. O professor indica explicitamente as saídas desejadas de modo a direcionar o treinamento, dessa forma é

possível medir o desempenho da rede através do sinal de erro, que é utilizado no ajuste dos pesos, já que o ajuste é feito com o objetivo de minimizar o erro.

Obviamente existem várias maneiras de ajustar os pesos da rede, isso é definido pelo algoritmo de treinamento escolhido. Existem muitos algoritmos para o treinamento de redes neurais. Eles diferem no modo de ajuste dos pesos, na rapidez de convergência, na garantia de convergência e na precisão.

O treinamento supervisionado é representado na Figura 13 abaixo:

Figura 13 – Treinamento supervisionado



3.3.2 Treinamento não supervisionado

Outro tipo de treinamento é o treinamento não supervisionado. Nesse tipo de treinamento não existe a figura do agente externo que direciona o treinamento, a rede dispõe apenas dos padrões de entrada. O objetivo da rede nesse tipo de treinamento é detectar padrões, semelhanças, regularidades entre os dados para agrupá-los. As semelhanças entre os dados são detectadas pelos neurônios.

Durante o processo de treinamento, as entradas são apresentadas à rede, e o neurônio que tiver o vetor de pesos mais próximo (distância euclidiana) do vetor de entrada tem os seus pesos ajustados de modo a ficar ainda mais próximo do vetor de entrada. A tendência é que com o tempo, neurônios ou conjunto de neurônios identifiquem grupos de dados que possuam características semelhantes (HAYKIN, 1999).

Esse tipo de treinamento é adequado quando o objetivo é agrupar ou classificar dados. Após encerrada a fase de treinamento, uma nova entrada ativará um neurônio ou grupo específico de neurônios que indicará a qual grupo ou classe a nova entrada inserida pertence. Um tipo de rede neural que utiliza treinamento não supervisionado são os mapas auto-organizáveis de Kohonen que realizam a tarefa de organizar dados em grupos.

3.3.3 Algoritmo de treinamento *backpropagation*

O *backpropagation* é um algoritmo que faz a correção do erro gerado pela rede utilizado no treinamento supervisionado. Primeiramente um padrão de entrada é apresentado à rede, o processamento ocorre, e então o erro é computado. Durante essa fase de propagação direta os pesos estão fixados. Após a propagação direta, o sinal de erro é retro-propagado pela rede na direção contrária dos pesos sinápticos, até a camada de entrada. Os pesos são então ajustados de forma a aproximar a resposta da rede da resposta desejada.

O ajuste nos pesos sinápticos é feito de modo a minimizar uma função de custo. No algoritmo *backpropagation* isso é feito de forma iterativa, a variação de um peso qualquer na rede é calculada com base na derivada da função custo em relação a esse peso, e numa *taxa de aprendizagem* definida pelo usuário. Assim pode-se definir a forma como é feito o ajuste nos pesos segundo o algoritmo pelas equações abaixo:

$$\Delta x_{ij}(n) = -\eta \frac{\partial E(n)}{\partial w_{ij}(n)} \quad (3.2)$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \quad (3.3)$$

x_{ij} - Peso sináptico da rede que liga o neurônio j ao neurônio i

η - Taxa de aprendizagem

E - Erro total gerado pela rede

n - Iteração corrente

O sinal negativo na Equação 3.2 justifica-se pois o gradiente indica a direção de crescimento local da função de custo, como queremos minimizá-la, devemos seguir a direção contrária à do gradiente.

Vale ressaltar que o esquema de ajuste definido na Equação 2 aproxima-se do popular método do gradiente da programação matemática. Já na literatura conexionista o cálculo das derivadas parciais do erro em relação aos pesos da rede e o emprego da Equação 2 são em conjunto denominados de método da retropropagação do erro (*Backpropagation*)

Precisa-se definir a função de custo para que o gradiente possa ser calculado. Para chegar à função de custo, primeiramente definimos o sinal de erro, que é medido nos neurônios que pertencem à camada de saída. A equação abaixo define o erro no neurônio de saída k na iteração n .

$$e_k(n) = d_k(n) - y_k(n) \quad (3.4)$$

Onde $d_k(n)$ é o valor de saída desejado (informado pelo “professor”) e $y_k(n)$ é o valor efetivamente calculado pela rede, ambos na iteração n .

Pode-se definir o erro total gerado pela rede na iteração n como sendo a soma dos erros quadráticos gerados por cada neurônio de saída, em que K é o número de neurônios de saída:

$$E(n) = \frac{1}{2} \sum_{k=1}^K e_k^2(n) \quad (3.5)$$

A função de custo que deve ser minimizada é o erro total médio, ou seja, a média dos erros totais gerados para cada padrão de treinamento. A função de custo está definida na Equação 3.6, onde N é o número de padrões usados no treinamento:

$$J = \frac{1}{N} \sum_{n=1}^N E(n) \quad (3.6)$$

É importante notar que minimizando o erro total estamos minimizando a função de custo. Essa função depende dos valores dos pesos da rede. Para tornar a dependência explícita vamos escrever as Equações da propagação direta da rede implementada neste trabalho. A rede possui uma camada oculta. Primeiramente o cálculo das saídas é feito de acordo com a Equação abaixo:

$$y_k(n) = f(s_k(n)) \quad (3.7)$$

O sinal $s_k(n)$ é a soma ponderada dos sinais que chegam ao neurônio de saída k.

$$s_k(n) = \sum_{q=0}^H v_{kq}(n) \cdot h_q(n) \quad (3.8)$$

Em que $h_0(n) = 1$; uma entrada constante para os neurônios de saída chamada de *bias*; $v_{kq}(n)$ é um peso da camada de saída da rede e o sinal $h_q(n)$ é calculado pela equação abaixo:

$$h_q(n) = f(r_q(n)) \quad (3.9)$$

O sinal $r_q(n)$ é a soma ponderada dos sinais que chegam ao neurônio oculto q:

$$r_q(n) = \sum_{m=0}^M u_{qm}(n) \cdot x_m(n) \quad (3.10)$$

Em que $x_0(n) = 1$; uma entrada constante para os neurônios ocultos chamada de *bias*; $u_{qm}(n)$ é um peso da camada de entrada da rede e $x_m(n)$ é uma entrada externa.

Utilizando as Equações 3.8, 3.9 e 3.10 pode-se deduzir facilmente as expressões para o gradiente do erro em relação aos pesos da camada de saída e da camada oculta. As Equações 3.11 e 3.12 mostram o resultado obtido:

$$\frac{\partial E(n)}{\partial v_{al}(n)} = -e_a(n) f'(s_a(n)) h_l(n) \quad (3.11)$$

$$\frac{\partial E(n)}{\partial u_{bp}(n)} = -f'(r_b(n)) x_p(n) \sum_{j=1}^C e_j(n) f'(s_j(n)) v_{jb}(n) \quad (3.12)$$

Em que $v_{al}(n)$ é um peso qualquer da camada de saída da rede e $u_{bp}(n)$ é um peso qualquer da camada oculta.

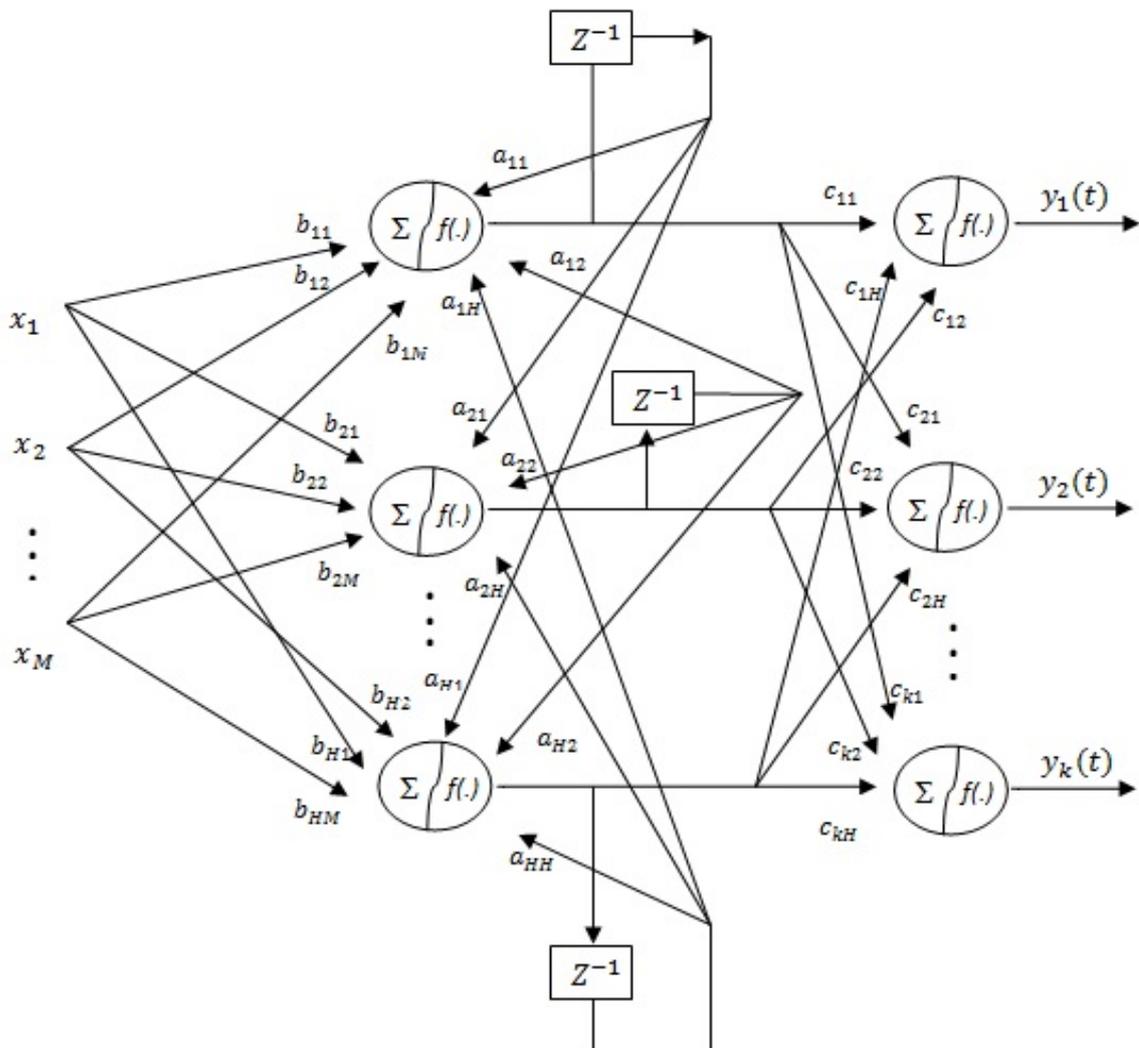
3.3.4 *Backpropagation* para as redes recorrentes

O treinamento supervisionado de redes recorrentes acontece da mesma forma que o treinamento de redes estáticas. Um agente externo direciona o treinamento fornecendo as saídas desejadas para que o sinal de erro possa ser calculado. O sinal de erro é então usado no ajuste dos pesos da rede. O treinamento pode ser feito inclusive utilizando o mesmo algoritmo *backpropagation*, adaptando porém, o cálculo da derivada do erro em relação aos pesos da rede.

Como a rede recorrente possui uma topologia diferente, já que existem conexões de realimentação, o cálculo do gradiente do erro em relação aos pesos é diferente. Nesse

trabalho foi implementada uma rede neural recorrente com a topologia que pode ser observada na Figura 14:

Figura 14 – Arquitetura da rede neural recorrente



Os pesos da rede foram divididos em 3 classes:

- Pesos classe ‘c’ ligam os neurônios da camada intermediária aos neurônios da camada de saída;
- Pesos classe ‘a’ são pesos de realimentação, ou seja, ligam os neurônios da camada intermediária aos próprios neurônios dessa camada;
- Pesos classe ‘b’ ligam as entradas externas aos neurônios da camada intermediária.

A rede recorrente implementada possui genericamente:

- C saídas;

- H neurônios ocultos (neurônios de camada intermediária);
- M entradas externas.

A função custo definida aqui é a mesma que é usada para as redes MLP, assim como o erro total da rede e o sinal de erro em cada saída da rede, foram definidos nas Equações 3.4, 3.5 e 3.6:

$$e_k(n) = d_k(n) - y_k(n)$$

$$E(n) = \frac{1}{2} \sum_{k=1}^K e_k^2(n)$$

$$J = \frac{1}{N} \sum_{n=1}^N E(n)$$

As saídas da rede são calculadas como apresentado a seguir:

$$y_k(n) = f(v_k(n)) \quad (3.13)$$

v_k é a soma dos sinais que chegam ao neurônio de saída k:

$$v_k(n) = \sum_{i=0}^H c_{ki}(n) \cdot u_i(n) \quad (3.14)$$

Na equação $c_{k0}(n)$ representa o *bias* já mencionado anteriormente, $u_i(n)$ representa os sinais de saída dos neurônios ocultos e é calculado da seguinte forma:

$$u_i(n) = f(h_i(n)) \quad (3.15)$$

$h_i(n)$ é a soma dos sinais que chegam ao neurônio oculto i:

$$h_i(n) = \sum_{j=1}^H a_{ij}(n) u_j(n-1) + \sum_{m=0}^M b_{im}(n) x_m(n) \quad (3.16)$$

Em que $b_{i0}(n)$ é o *bias*. Assim como nas redes diretas, o ajuste dos pesos para as redes recorrentes é feito de acordo com as equações 3.2 e 3.3. Para poder utilizá-las, é necessário calcular a derivada do erro total em relação a cada classe de peso. A seguir, é feito o cálculo do gradiente para cada tipo de peso separadamente.

Pesos da classe ‘c’:

Cálculo de $\frac{\partial E(n)}{\partial c_{ri}(n)}$ onde c_{ri} representa um peso qualquer da classe c:

$$\frac{\partial E(n)}{\partial c_{ri}(n)} = \frac{\partial(\frac{1}{2} \sum_{k=1}^C e_k^2(n))}{\partial c_{ri}(n)}$$

O peso $c_{ri}(n)$ só influencia na saída $y_r(n)$ e, portanto, apenas no erro $e_r(n)$:

$$\frac{\partial E(n)}{\partial c_{ri}(n)} = \frac{\partial E(n)}{\partial e_r(n)} \frac{de_r(n)}{dy_r(n)} \frac{dy_r(n)}{dv_r(n)} \frac{\partial v_r(n)}{\partial c_{ri}(n)}$$

De acordo com a equação 3.5:

$$\frac{\partial E(n)}{\partial e_r(n)} = e_r(n)$$

Utilizando a equação 3.4:

$$\frac{de_i(n)}{dy_i(n)} = -1$$

De acordo com a equação 3.13:

$$\frac{dy_r(n)}{dv_r(n)} = f'(v_r(n))$$

E pela equação 3.14:

$$\frac{\partial v_r(n)}{\partial c_{ri}(n)} = u_i(n)$$

Dessa forma, o gradiente do erro total em relação a um peso qualquer da classe c pode ser escrito da seguinte forma:

$$\frac{\partial E(n)}{\partial c_{ri}(n)} = -e_r(n) f'(v_r(n)) u_i(n) \quad (3.17)$$

Pesos da classe b:

Cálculo de $\frac{\partial E(n)}{\partial b_{sq}(n)}$ onde b_{sq} representa um peso qualquer da classe b

$$\frac{\partial E(n)}{\partial b_{sq}(n)} = \frac{\partial(\frac{1}{2} \sum_{k=1}^C e_k^2(n))}{\partial b_{sq}(n)}$$

$$\frac{\partial E(n)}{\partial b_{sq}(n)} = \sum_{k=1}^C e_k(n) \frac{\partial y_k(n)}{\partial b_{sq}(n)} \quad (3.18)$$

Cálculo de $\frac{\partial y_k(n)}{\partial b_{sq}(n)}$:

$$\frac{\partial y_k(n)}{\partial b_{sq}(n)} = \frac{dy_k(n)}{dv_k(n)} \frac{\partial v_k(n)}{\partial b_{sq}(n)}$$

Utilizando a equação 3.13:

$$\frac{\partial y_k(n)}{\partial b_{sq}(n)} = f'v_k(n) \frac{\partial v_k(n)}{\partial b_{sq}(n)} \quad (3.19)$$

Cálculo de $\frac{\partial v_k(n)}{\partial b_{sq}(n)}$:

$$\frac{\partial v_k(n)}{\partial b_{sq}(n)} = \frac{\partial \sum_{i=1}^H c_{ki}(n)u_i(n)}{\partial b_{sq}(n)}$$

$$\frac{\partial v_k(n)}{\partial b_{sq}(n)} = \sum_{i=1}^H c_{ki}(n) \frac{\partial u_i(n)}{\partial b_{sq}(n)} \quad (3.20)$$

Cálculo de $\frac{\partial u_i(n)}{\partial b_{sq}(n)}$:

$$\frac{\partial u_i(n)}{\partial b_{sq}(n)} = \frac{du_i(n)}{dh_i(n)} \frac{\partial h_i(n)}{\partial b_{sq}(n)}$$

Utilizando a equação 3.15:

$$\frac{\partial u_i(n)}{\partial b_{sq}(n)} = f'(h_i(n)) \frac{\partial h_i(n)}{\partial b_{sq}(n)} \quad (3.21)$$

Cálculo de $\frac{\partial h_i(n)}{\partial b_{sq}(n)}$:

$$\frac{\partial h_i(n)}{\partial b_{sq}(n)} = \frac{\partial (\sum_{j=1}^H a_{ij}(n)u_j(n-1) + \sum_{m=0}^M b_{im}(n)x_m(n))}{\partial b_{sq}(n)}$$

$$\frac{\partial h_i(n)}{\partial b_{sq}(n)} = \begin{cases} \sum_{j=1}^H a_{sj}(n) \frac{\partial u_j(n-1)}{\partial b_{sq}(n)} + x_q(n) & s = i \\ \sum_{j=1}^H a_{ij}(n) \frac{\partial u_j(n-1)}{\partial b_{sq}(n)} & s \neq i \end{cases} \quad (3.22)$$

No encadeamento das Equações 3.18, 3.19, 3.20, 3.21e 3.22 pode-se calcular o gradiente do erro total em relação a um peso qualquer da classe \mathbf{b} . A expressão completa não é escrita neste trabalho pelo fato da mesma ser muito extensa.

Pesos classe a:

Cálculo de $\frac{\partial E(n)}{\partial a_{lp}(n)}$ onde $a_{lp}(n)$ representa um peso qualquer da classe a :

$$\frac{\partial E(n)}{\partial a_{lp}(n)} = \frac{\partial(\frac{1}{2} \sum_{k=1}^C e_k^2(n))}{\partial a_{lp}(n)}$$

$$\frac{\partial E(n)}{\partial a_{lp}(n)} = \sum_{k=1}^C e_k(n) \frac{\partial y_k(n)}{\partial a_{lp}(n)} \quad (3.23)$$

Cálculo de $\frac{\partial y_k(n)}{\partial a_{lp}(n)}$:

$$\frac{\partial y_k(n)}{\partial a_{lp}(n)} = \frac{dy_k(n)}{dv_k(n)} \frac{\partial v_k(n)}{\partial a_{lp}(n)}$$

Utilizando a equação 3.13:

$$\frac{\partial y_k(n)}{\partial a_{lp}(n)} = f'_{v_k(n)} \frac{\partial v_k(n)}{\partial a_{lp}(n)} \quad (3.24)$$

Cálculo de $\frac{\partial v_k(n)}{\partial a_{lp}(n)}$:

$$\frac{\partial v_k(n)}{\partial a_{lp}(n)} = \frac{\partial \sum_{i=1}^H c_{ki}(n) u_i(n)}{\partial a_{lp}(n)}$$

$$\frac{\partial v_k(n)}{\partial a_{lp}(n)} = \sum_{i=1}^H c_{ki}(n) \frac{\partial u_i(n)}{\partial a_{lp}(n)} \quad (3.25)$$

Cálculo de $\frac{\partial u_i(n)}{\partial a_{lp}(n)}$:

$$\frac{\partial u_i(n)}{\partial a_{lp}(n)} = \frac{du_i(n)}{dh_i(n)} \frac{\partial h_i(n)}{\partial a_{lp}(n)}$$

Cálculo de $\frac{\partial h_i(n)}{\partial a_{lp}(n)}$:

$$\frac{\partial h_i(n)}{\partial a_{lp}(n)} = f'(h_i(n)) \frac{\partial h_i(n)}{\partial a_{lp}(n)} \quad (3.26)$$

Cálculo de $\frac{\partial h_i(n)}{\partial a_{lp}(n)}$:

$$\frac{\partial h_i(n)}{\partial a_{lp}(n)} = \frac{\partial(\sum_{j=1}^H a_{ij}(n)u_j(n-1) + \sum_{m=0}^M b_{im}(n)x_m(n))}{\partial a_{lp}(n)}$$

$$\frac{\partial h_i(n)}{\partial a_{lp}(n)} = x = \begin{cases} \sum_{j=1}^H a_{ij}(n) \frac{\partial u_j(n-1)}{\partial a_{lp}(n)} + u_p(n-1) & l = i \\ \sum_{j=1}^H a_{ij}(n) \frac{\partial u_j(n-1)}{\partial a_{lp}(n)} & l \neq i \end{cases} \quad (3.27)$$

No encadeamento das Equações 3.23, 3.24, 3.25, 3.26 e 3.27 pode-se calcular o gradiente do erro total em relação a um peso qualquer da classe \mathbf{a} , mas a expressão completa não é escrita neste trabalho pelo fato da mesma ser muito extensa.

Uma importante observação a ser feita é que nas equações para o cálculo do gradiente para os pesos das classes a e b deve-se calcular derivadas da forma $\frac{\partial u_j(n-1)}{\partial x(n)}$, como não existe expressão algébrica para o cálculo dessa derivada, utilizou-se uma aproximação matemática que é válida se considerarmos que o ajuste nos pesos é muito pequeno. A aproximação utilizada é a seguinte:

$$\frac{\partial u_j(n-1)}{\partial x(n)} \approx \frac{\partial u_j(n-1)}{\partial x(n-1)}$$

Considerando essa aproximação, as derivadas podem ser calculadas de forma recorrente.

3.3.5 O algoritmo Rprop

O algoritmo *Backpropagation* por ser um método de primeira ordem e usar uma taxa de aprendizagem fixa, e por depender da magnitude do gradiente, tipicamente demanda um tempo longo para que haja convergência e pode apresentar comportamento oscilatório quando próximo de um ótimo local (HAYKIN, 1999).

Neste trabalho optou-se por utilizar o algoritmo Rprop, que é um método de aprendizado de eficiência já comprovada na literatura, e dentre os de primeira ordem é um dos que apresenta convergência mais rápida.

Diferentemente do *backpropagation*, o Rprop não utiliza a magnitude do gradiente para fazer o ajuste nos pesos da rede, é necessário apenas o seu sinal. Algumas vantagens do algoritmo Rprop citadas por C.Igel e Husk (2003) e Riedmiller (1994) são descritas abaixo:

- O Rprop é robusto em relação aos seus parâmetros internos, além disso, os parâmetros são intuitivos e de fácil ajuste;
- O Rprop é um algoritmo de primeira ordem e a complexidade varia apenas linearmente com o número de parâmetros que precisam ser otimizados;
- O Rprop é um algoritmo geral, funciona bem não importando as propriedades da topologia da rede em questão;
- O ajuste dos pesos depende somente do sinal do gradiente, evitando problemas que podem ocorrer quando a magnitude do gradiente é muito baixa ou muito elevada, como saltos sobre mínimos locais e convergência lenta por ajuste pequeno nos pesos;
- Fácil implementação.

3.3.5.1 Descrição do algoritmo

O Rprop ajusta os pesos da rede somando a eles um valor de atualização $\Delta\theta_i^{(t)}$ que é calculado a cada iteração e para cada peso. Para determinar a direção do ajuste, é utilizado apenas o sinal do gradiente em batelada, que é o gradiente somado sobre todos os padrões de treinamento.

$$\Delta\theta_i^{(t+1)} = \theta_i^{(t)} + \Delta\theta_i^{(t)} \quad (3.28)$$

$$\Delta\theta_i^{(t)} = \begin{cases} -\Delta_i^{(t)} & \frac{\partial E^{(t)}}{\partial \theta_i} > 0 \\ +\Delta_i^{(t)} & \frac{\partial E^{(t)}}{\partial \theta_i} < 0 \\ 0 & \frac{\partial E^{(t)}}{\partial \theta_i} = 0 \end{cases} \quad (3.29)$$

θ_i : Um peso qualquer de uma rede direta ou recorrente.

$\Delta\theta_i^{(t)}$: Valor acrescido ao peso θ_i na iteração t

$\Delta_i^{(t)}$: Valor absoluto do acréscimo ao peso. ($\Delta\theta_i^{(t)}$, a menos de um sinal)

$\frac{\partial E^{(t)}}{\partial \theta_i}$: Gradiente somado sobre todos os padrões de treino na iteração t .

O valor absoluto do acréscimo é modificado automaticamente com base no comportamento do sinal do gradiente ao longo das iterações. Uma mudança no sinal do gradiente de uma iteração para outra, indica que a atualização dos pesos foi muito grande e o algoritmo ‘saltou’ um mínimo local. Portanto, devemos diminuir o módulo do acréscimo dos pesos. Se o sinal não sofre mudança, pode-se aumentar o módulo valor do acréscimo na

esperança de acelerar o processo de convergência em regiões em que o erro está diminuindo. Esta idéia é expressa matematicamente por:

$$\Delta_i^{(t)} = \begin{cases} \mu^+ \cdot \Delta_i^{(t-1)} & \frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i} > 0 \\ \mu^- \cdot \Delta_i^{(t-1)} & \frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i} < 0 \\ \Delta_i^{(t)} & \frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i} = 0 \end{cases} \quad (3.30)$$

Em que $0 < \mu^- < 1 < \mu^+$. Esse é o Rprop original proposto por Riedmiller e Braun. No presente trabalho utiliza-se uma versão melhorada do algoritmo proposta em C.Igel e Husk (2003). A principal modificação ocorre na mudança da estrutura de decisão quando ocorre mudança no sinal do gradiente acumulado. Uma mudança no sinal do gradiente indica que o algoritmo ‘saltou’ um mínimo local. Caso isso aconteça, é feito um teste no erro, se houver aumento, o ajuste anterior dos pesos é revertido (*weigh backtracking*), se houver decréscimo o algoritmo segue normalmente. A estrutura de decisão modificada para o ajuste do acréscimo é apresentada abaixo:

Se $\frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i} < 0$ então:

$$\Delta_i^{(t)} = \begin{cases} -\Delta \theta_i^{t-1} & E^{(t)} > E^{(t-1)} \\ 0 & E^{(t)} < E^{(t-1)} \end{cases} \quad (3.31)$$

O pseudocódigo do algoritmo Rprop completo com as modificações está escrito abaixo:

Para cada w_{ij} {

Se $\frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i} > 0$ então {

$$\Delta_i^{(t)} = \min(\Delta_i^{(t-1)} \eta^+, \Delta_{max})$$

$$\Delta \theta_i^{(t)} = -\text{sign}\left(\frac{\partial E^{(t)}}{\partial \theta_i}\right) \Delta_i^{(t)}$$

$$\Delta \theta_i^{(t+1)} = \theta_i^{(t)} + \Delta \theta_i^{(t)}$$

}

Senão, se $\frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i} < 0$ então {

$$\Delta_i^{(t)} = \max(\Delta_i^{(t-1)} \eta^-, \Delta_{min})$$

$$\text{se } E^{(t)} > E^{(t-1)} \text{ então } \Delta \theta_i^{(t+1)} = \theta_i^{(t)} - \Delta \theta_i^{(t)}$$

$$\frac{\partial E^{(t)}}{\partial \theta_i} = 0$$

$$\begin{aligned}
& \} \\
& \text{senão } \{ \\
& \Delta\theta_i^{(t)} = -\text{sign}\left(\frac{\partial E^{(t)}}{\partial \theta_i}\right)\Delta_i^{(t)} \\
& \qquad \qquad \qquad \Delta\theta_i^{(t+1)} = \theta_i^{(t)} + \Delta\theta_i^{(t)} \qquad (3.32) \\
& \} \\
& \}
\end{aligned}$$

As variáveis Δ_{max} e Δ_{min} definem o valor máximo e o valor mínimo que o módulo do acréscimo pode atingir e são utilizadas para controle.

A instrução $\frac{\partial E^{(t)}}{\partial \theta_i} = 0$ é utilizada quando há mudança no sinal do gradiente. Essa instrução garante que, no passo seguinte, o módulo do acréscimo não será aumentado, pois ativará o último ‘senão’, já que o produto $\frac{\partial E^{(t)}}{\partial \theta_i} \frac{\partial E^{(t-1)}}{\partial \theta_i}$ será igual a zero.

Apesar das vantagens apresentadas no início desta sessão, uma importante informação sobre o Rprop é que ele ajusta os pesos da rede de forma separada, ou seja, cada peso é ajustado considerando todos os outros constantes. O Rprop procura otimizar a função de custo ao longo de cada dimensão. O algoritmo considera uma hipótese de risco e não oferece garantia de ajustes minimizantes.

Na próxima sessão é apresentado o algoritmo de parada antecipada, que ameniza o problema do sobre ajuste, que pode ocorrer no treinamento de redes neurais e que também é explicado na sessão seguinte.

3.3.6 Algoritmo de parada antecipada

Como resultado do processo de treinamento, deseja-se que a rede neural generalize bem, ou seja, que também produza saídas apropriadas para novas entradas que respeitem a distribuição de probabilidade que caracteriza o conjunto de treinamento.

A capacidade de generalização da rede pode ser prejudicada por um fenômeno chamado de sobre-ajuste (*over fitting*) que acontece quando a rede é treinada excessivamente em um conjunto de dados e passa a assimilar as irregularidades ou ruído contidos no conjunto de treinamento. Durante o treino o erro diminui, porém, em certo momento o erro para dados não vistos na fase de treinamento começa a aumentar. O que ocorre é que a rede fica “especializada” demais nos exemplos de treino e começa a divergir, perder capacidade de generalização, para outros exemplos. Isso acontece porque geralmente a rede possui muitos parâmetros, tornando o modelo muito flexível.

As duas principais abordagens para amenizar o problema do sobre-ajuste são diminuir o número de parâmetros, e a parada antecipada. A parada antecipada tem

seu desempenho superior à situação em que nenhum controle de sobre-ajuste é empregado (PRECHELT, 1998).

De maneira simplificada, a parada antecipada consiste em treinar a rede sobre o conjunto de treinamento e, ao mesmo tempo, avaliar o erro em outro conjunto de dados mantido à parte do ajuste de pesos (erro de validação). Quando o erro de validação começar a aumentar deve-se parar o treinamento e utilizar o conjunto de pesos que apresentou o menor erro de validação, já que a validação está medindo a capacidade de generalização da rede.

Esta explanação assume que o erro de validação apresenta um comportamento ideal, como o apresentado na Figura 15. Porém o comportamento real do erro de validação é bem mais complexo, muitas vezes apresentando mais de um mínimo local, como por exemplo, o comportamento mostrado na Figura 16. Dessa forma é difícil saber quando o mínimo global foi atingido.

Figura 15 – Erro de validação ideal x erro de treino ao longo das épocas

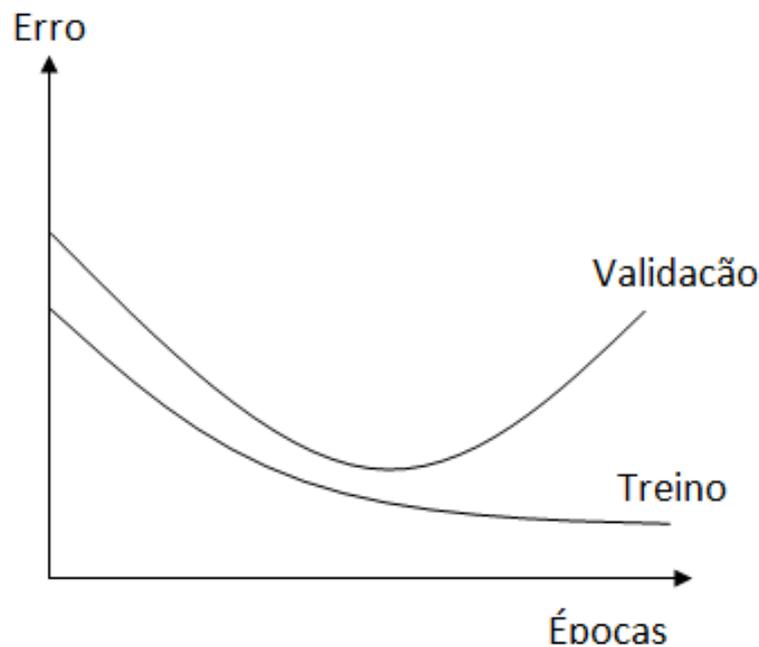
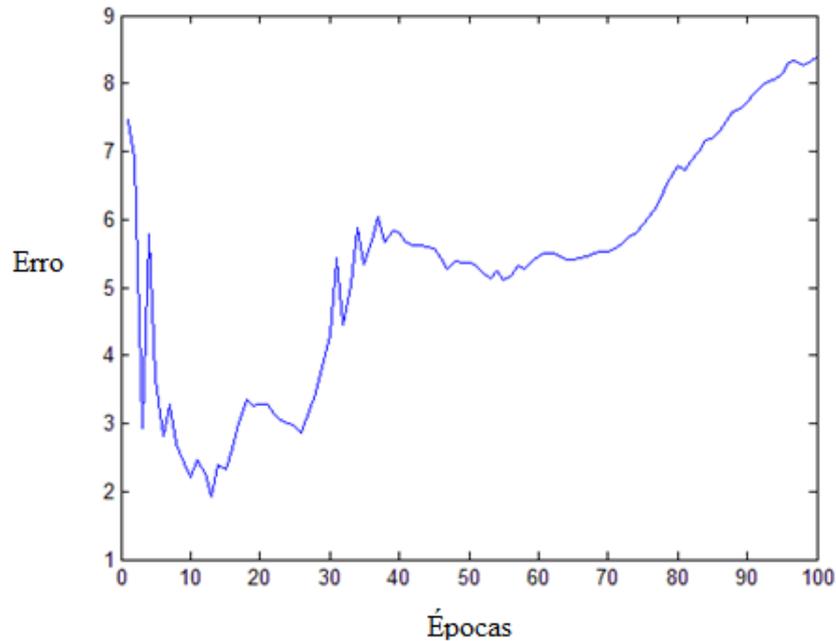


Figura 16 – Erro de validação real



Para lidar com as oscilações presentes nas curvas de erro de validação, Prechelt (1998) propõe uma heurística denominada de parada antecipada que funciona da seguinte maneira:

- O erro de validação é avaliado a cada k épocas;
- Se o erro de validação aumentar após s avaliações sucessivas, deve-se parar o treino e recuperar o conjunto de pesos que apresentou o menor erro até então.

Esta heurística assume que se o erro de validação aumentar em s checagens sucessivas, o sobre ajuste final foi atingido, e a partir deste momento o erro de validação tende só a aumentar. Obviamente essa afirmação não é sempre verdadeira. A escolha dos parâmetros k e s influencia bastante.

O próximo capítulo apresenta a metodologia de solução utilizada, arquitetura da rede e todos os parâmetros escolhidos para o ambiente computacional.

4 METODOLOGIA

De posse da série temporal PLD e das séries temporais das variáveis externas (explanadas mais à frente), foi feita uma análise de desempenho das técnicas na tarefa de previsão da série PLD um passo a frente. As técnicas utilizadas são as redes neurais diretas e recorrentes além da técnica ARIMA (*Auto regressive integrated moving average*), um modelo muito utilizado para previsão de séries temporais e que pode servir de base de comparação de desempenho.

Antes de proceder à metodologia propriamente dita, é feita uma breve explanação sobre o modelo ARIMA.

4.1 O modelo ARIMA

O modelo ARIMA é uma generalização dos modelos auto-regressivos, que apresentam a característica de relacionar dados num certo instante com dados em instantes passados, ou seja, dados são calculados com base em dados passados. O modelo ARIMA apresenta, além da parte auto-regressiva (AR), o componente de média móvel (MA), indicando que os termos de erro são combinação linear de erros ocorridos em dados passados, e o componente de integração (I) que indica o processo de diferenciação entre os dados atuais e dados passados.

Os modelos ARIMA são geralmente denotados por $ARIMA(p, d, q)$ em que p , d e q são inteiros não negativos. P é a ordem (número de lags) do modelo auto-regressivo, d é o grau de diferenciação e q é a ordem da média móvel.

O modelo é representado matematicamente da seguinte maneira:

$$y_t = a_1 y_{t-1} + \dots + a_p y_{t-p} + e_t + b_1 e_{t-1} + \dots + b_q e_{t-q} \quad (4.1)$$

Em que o índice t representa o instante da amostra, os parâmetros do modelo p e q representam a ordem do modelo auto-regressivo e de média móvel. O parâmetro d não é aparente na equação pois refere-se à multiplicidade de raízes do polinômio. Os a e b são coeficientes constantes do modelo auto-regressivo e de média móvel respectivamente.

4.2 Aspectos gerais

Nesta seção trata-se das diferentes situações de simulação, as entradas definidas para as redes neurais, as configurações da máquina na qual as simulações foram realizadas e a versão do software utilizado.

Deve-se escolher as variáveis de entrada para as redes neurais entre as cinco que estão disponíveis. Essa escolha foi feita com base em correlação como será explanado adiante. Além da seleção das variáveis de entrada, deve-se escolher também a quantidade de dados e atrasos no tempo para cada uma delas, por exemplo, para uma variável de entrada deve-se utilizar dados atuais? Dados de uma semana atrás ou combinar atuais e passados? Deve-se selecionar quais lags(atrasos nas entradas) serão utilizados como entrada em cada variável externa.

No caso das redes neurais diretas, houve duas situações de simulação: Primeiramente são utilizados, para cada variável de entrada, os 'n' primeiros lags, variando n de 1 à 10. Depois um algoritmo é usado para selecionar os lags mais significativos para cada variável de entrada. Já no caso das redes recorrentes a escolha de lags não é necessária visto que a natureza da rede já leva em consideração os atrasos na entrada.

A seleção das variáveis de entrada é utilizada para todas as técnicas aplicadas, já a seleção de lags somente para a rede neural direta. Tem-se 4 opções para o ataque ao problema cujos desempenhos serão avaliados

- 1) Modelo ARIMA
- 2) Rede neural recorrente
- 3) Rede neural direta com 'n' primeiros lags
- 4) Rede neural direta com seleção de lags

As simulações foram executadas em um notebook intel(R) core(TM) com CPU de 1.80 GHz e 8 GB de memória RAM com sistema operacional Windows 8. O programa utilizado foi o matlab versão 8.1.0.604.

4.3 Dados

A série temporal de preços PLD foi coletada junto à página da CCEE na internet (<http://www.ccee.org.br/>) indo de 04/08/2001 26/12/2009. A série apresenta os valores de PLD semanais para os 4 sub-mercados: Norte, Nordeste, Sul e Sudeste/Centro-Oeste, em três patamares: pesado, médio e leve. Ao todo, há 440 padrões de dados disponíveis nesta série.

Além da série alvo (PLD), foram consideradas 5 variáveis externas (exógenas) como entradas para as redes neurais: Geração hidráulica média (GHID), Geração térmica média (GTER), Carga de energia média (Q), Quantidade de água armazenada e reservada em reservatório transformada em energia (EARM) e Quantidade de água que entra em reservatório transformada em energia (ENA). As séries temporais correspondentes a estas

variáveis foram obtidas no site do ONS (<http://www.ons.org.br/publicacao/ipdo>). As três primeiras variáveis são medidas em megawatts e as outras duas são porcentagens. Estas variáveis se assemelham ao conjunto de entradas utilizadas pelos softwares de otimização do ONS. O conjunto de dados disponível para estas séries é maior que o disponível para a série PLD, portanto, houve pré-processamento dos dados para igualar o período de tempo representado e o intervalo de amostragem de todas as séries.

4.4 Seleção de entradas

A escolha das variáveis externas para cada um dos submercados é feita com base nos resultados da correlação entre cada entrada e a saída (PLD), levando-se em conta os patamares de a carga Pesado, Médio e Leve. As Tabelas 1, 2, 3 e 4 mostram estas medidas de correlação:

Tabela 1 – Correlação para o mercado Sudeste/Centro-Oeste

	Pesado	Médio	Leve
GHID	-0,2798	-0,2825	-0,2862
GTER	-0,2389	-0,2363	-0,2328
Q	-0,1951	-0,1977	-0,2013
EARM	-0,2102	-0,2106	-0,2109
ENA	-0,2492	-0,2499	-0,2514

Tabela 2 – Correlação para o submercado Norte

	Pesado	Médio	Leve
GHID	-0,1514	-0,1520	-0,1681
GTER	Não disponível	Não disponível	Não disponível
Q	-0,1434	-0,1453	-0,1519
EARM	-0,0379	-0,0377	-0,0373
ENA	-0,2235	-0,2254	-0,2287

Tabela 3 – Correlação para o submercado Nordeste

	Pesado	Médio	Leve
GHID	0,4667	0,4674	0,4686
GTER	0,6374	0,6382	0,6390
Carga	0,5042	0,5048	0,5059
EARM	-0,1169	-0,1169	-0,1167
ENA	-0,2567	-0,2565	-0,2551

Tabela 4 – Correlação para o submercado Sul

	Pesado	Médio	Leve
GHID	0,0526	0,0549	0,0714
GTER	0,2894	0,2903	0,2984
Carga	0,2294	0,2378	0,2279
EARM	-0,1038	-0,1029	-0,0950
ENA	-0,0965	-0,0975	-0,0951

É importante notar que, em cada submercado, as variáveis influenciam o PLD de forma diferente. O submercado nordeste, por exemplo, apresentou os maiores valores de correlação o que pode indicar uma facilidade maior ao prever o preço da energia utilizando essas variáveis. Para o submercado norte, não houve geração térmica no período analisado.

Para a previsão do PLD, todas as entradas externas foram utilizadas nos submercados Sudeste/Centro-Oeste e Nordeste em qualquer patamar. A variável quantidade de água armazenada transformada em energia (EARM) não foi utilizada na previsão da série para o submercado norte por apresentar baixa correlação (abaixo de 0,04), dessa forma não apresentando influencia significativa na saída. Pelo mesmo motivo, as variáveis geração hidráulica (GHID) e quantidade de água que entra em reservatório transformada em energia (ENA) não foram utilizadas para o submercado sul (correlações abaixo de 0,1).

4.5 Algoritmo de seleção de lags

A seleção dos lags consiste em determinar um limiar de correlação de modo a escolher apenas os lags s que apresentam correlação igual ou superior a este limiar. A correlação é aplicada atrasando as entradas ao longo de um ano, 52 pontos, já que a amostragem é semanal. Dessa forma tem-se uma função correlação no tempo, em que as

abscissas representam a variável de entrada e suas versões atrasadas $x(t)$, $x(t-1)$ etc, e as ordenadas são valores de correlação. O limiar é determinado como sendo uma porcentagem da correlação do lag mais significativo. Neste trabalho foram testados limiares de 90% e 95%. Logo, somente lags que possuem correlação próxima à do lag mais significativo serão selecionados como entrada em cada variável externa.

A escolha por selecionar dentre os 52 últimos pontos se dá por conta da sazonalidade anual das séries, com os valores de um ano sendo similares aos de outros anos (embora essa sazonalidade não seja sempre seguida e nem tenha precisão alta sempre). As Figuras 17 e 18 mostram o funcionamento do algoritmo de seleção:

Figura 17 – Seleção de lags: Variável externa Carga, no submercado Sul, patamar de carga pesado.

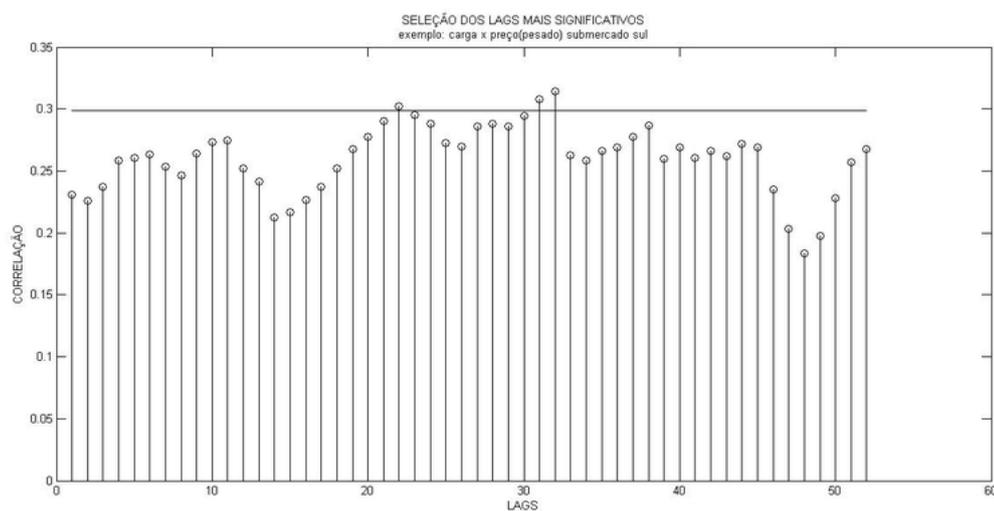
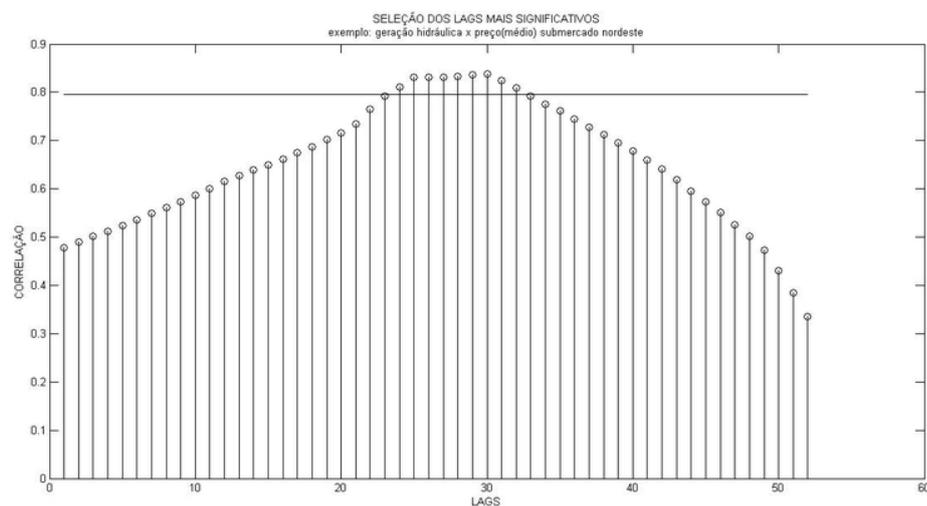


Figura 18 – Seleção de lags: Variável externa geração hidráulica, submercado Nordeste, patamar de carga médio.



Da Figura 17 tem-se que a variável externa Carga, para o patamar de carga pesado do submercado Sul, será utilizada nos instantes 21, 31 e 32 na entrada da rede direta. Da Figura 18 tem-se que a variável externa Geração Hidráulica, para a carga no patamar médio do submercado Nordeste, será utilizada nos instantes 24, 25, 26, 27, 28, 29, 30, 31 e 32 na entrada da rede direta.

As figuras 17 e 18 explicitam um fato importante: Nem sempre os lags mais significativos são os primeiros, ou seja, os que representam a informação mais recente. Em várias situações a informação de diversas semanas atrás é mais relevante que a informação recente para fazer a previsão de valores futuros do PLD.

Como foi explicado anteriormente, a seleção de lags não é utilizada na rede neural recorrente pois ela na sua natureza já considera a influência das entradas passadas.

4.6 Parametrização

Nesta seção serão explicadas todas as condições de simulação, os parâmetros e a discussão de suas escolhas.

4.6.1 Simulação

As simulações consideraram cada submercado de forma separada e para cada submercado os 3 patamares de carga (leve, médio e pesado) totalizando 12 combinações nas quais será testado o desempenho das técnicas escolhidas.

Foram utilizadas 100 condições iniciais (valores iniciais de pesos) para cada configuração de rede neural. A medida de desempenho usada para avaliar o desempenho das técnicas foi a média dos erros médios quadráticos normalizados, calculada sobre todas as condições iniciais para cada configuração de rede. A variável alvo e as variáveis externas foram normalizadas para valores entre 0 e 1. Os pesos de todas as redes neurais simuladas foram gerados aleatoriamente com valores entre 0 e 1. Os valores pequenos de pesos e entradas evita o problema de saturação na função de ativação dos neurônios.

4.6.2 Redes neurais e treinamento

O treinamento das redes neurais foi realizado em 500 épocas. Os padrões de dados utilizados no treinamento foram divididos em 60% para treino, 20% validação e 20% teste. (obs: a palavra 'treinamento está sendo usada no sentido amplo, referindo-se ao processo completo, já a palavra 'treino' está no sentido estrito, referindo-se a uma parte do processo). Esse tipo de escolha de épocas e divisão dos dados de treino é padrão e muito utilizado em simulações com redes neurais, a opção feita aqui seguiu um padrão comum.

As redes utilizadas no trabalho usaram o algoritmo rprop para o treinamento e também o algoritmo de parada antecipada para tratar o *over fitting*. Os parâmetros para

o algoritmo rprop foram:

$$\eta^+ = 1.2$$

$$\eta^- = 0.5$$

$$\Delta_{max} = 50$$

$$\Delta_{min} = 10^{-6}$$

A escolha desses parâmetros foi baseada na experiência dos autores de C.Igel e Husk (2003) que testaram vários valores de parâmetros e apresentaram essa combinação como sendo a que obteve os melhores resultados.

Os parâmetros para o algoritmo de parada antecipada foram:

$$k = 1$$

$$s = 501$$

Essa escolha de parâmetros foi feita visando a máxima eficiência do algoritmo de parada antecipada em troca de um pouco mais de custo computacional. Dessa forma o erro de validação será calculado em todas as épocas e o conjunto de pesos escolhido será o que apresentar o menor erro de validação em qualquer hipótese.

4.6.2.1 Discussão da quantidade de neurônios na camada oculta

A princípio poderia-se pensar que, para uma comparação justa, as redes direta e recorrente deveriam ser implementadas com o mesmo número de neurônios na camada oculta. Entretanto, caso esse critério seja utilizado, existe o favorecimento da rede neural direta. A razão é que se ambas apresentarem a mesma quantidade de neurônios, a rede recorrente será muito mais complexa computacionalmente e esse custo afetará drasticamente seu desempenho.

Suponha que uma rede neural recorrente possua N neurônios na camada oculta. Como cada um deles se conecta a si mesmo e a todos os outros, a rede terá a seguinte quantidade de pesos Q :

$$Q = N + (N - 1) + (N - 2) + \dots + 1 = \frac{N(N + 1)}{2} \quad (4.2)$$

Essa é a quantidade de parâmetros ajustáveis adicionais que a rede recorrente apresentará em relação à rede direta. Percebe-se que a complexidade varia com o quadrado de N , assim o acréscimo de um único neurônio nessa camada gera uma quantidade considerável de parâmetros.

Dessa forma o critério utilizado para a comparação foi estipular números diferentes de neurônios ocultos para os tipos de redes diferentes, permitindo que cada tipo alcance

seu melhor desempenho, pois o foco do trabalho é comparar a performance dos tipos de redes e não checar a influência do número de neurônios ocultos no resultado.

Para a rede neural direta, 16 neurônios foram usados na camada oculta, já para a rede neural recorrente, foram 6 neurônios. A escolha do número de neurônios foi baseada em simulações prévias mais sucintas, considerando menos condições iniciais. Esses testes anteriores foram realizados para determinar uma arquitetura de bom desempenho para cada tipo de rede utilizado. O menor número de neurônios escolhido para a rede recorrente é devido à sua alta complexidade e alto número de parâmetros. É esperado que o melhor desempenho da rede recorrente ocorra com um número menor de neurônios ocultos, pois isso leva a uma arquitetura com menos parâmetros ajustáveis. Em raciocínio análogo, faz sentido o maior número de neurônios ocultos para a rede direta ser maior em virtude da sua baixa complexidade e menor número de parâmetros ajustáveis.

O próximo capítulo apresenta os resultados das simulações realizadas.

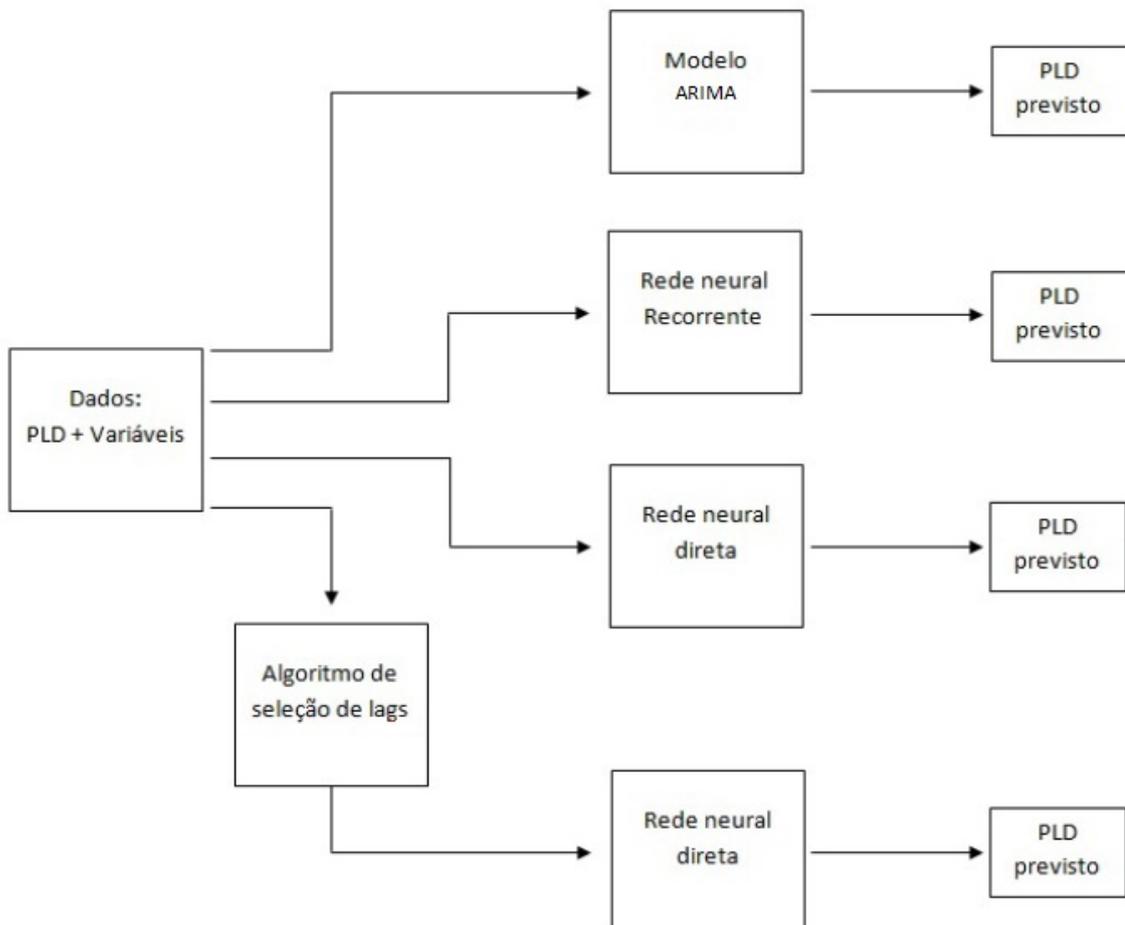
5 RESULTADOS DAS SIMULAÇÕES

Neste capítulo são mostrados os resultados de erro obtidos utilizando as redes neurais e o modelo ARIMA na tarefa de previsão do PLD um passo à frente (representado no caso, uma semana à frente). Os resultados são apresentados para cada submercado na seguinte ordem:

- 1) Modelo ARIMA
- 2) Rede neural recorrente
- 3) Rede neural direta sem seleção de lags
- 4) Rede neural direta com seleção de lags

A Figura 19 abaixo demonstra a utilização dos preditores mencionados na tarefa de previsão do PLD:

Figura 19 – Utilização dos preditores na previsão do PLD



O erro médio quadrático normalizado foi escolhido como forma de comparação de desempenho usada neste trabalho mas também são mostrados resultados de erros percentuais absolutos em relação ao resultado real.

O erro quadrático médio normalizado é calculado pela equação:

$$E_{quad} = \frac{1}{N} \sum_{i=1}^N (y - \hat{y})^2 \quad (5.1)$$

Em que N é o número de padrões de teste, y é o valor desejado real e \hat{y} é o valor previsto pela técnica. O erro é normalizado pois é calculado com as variáveis real e prevista normalizadas.

O erro médio percentual absoluto é calculado pela equação:

$$E_{percent} = \frac{1}{N} \sum_{i=1}^N \frac{(y - \hat{y})}{y} .100 \quad (5.2)$$

O erro percentual é mais intuitivo de perceber e por isso foi escolhido como segunda métrica avaliada.

A título de informação adicional, o modelo de referência ARIMA utilizado para cada submercado e cada patamar de carga foi o ARIMA(2,1,1). Esses foram os parâmetros determinados pelo matlab ao se inserir os dados no modelo. O modelo de referência encontrado foi o mesmo em todos os submercados e patamares de carga.

Devido a grande quantidade de resultados, escolheu-se exibi-los divididos por submercado. Dessa forma, a análise é facilitada e colocada de forma mais didática.

5.1 Submercado Sudeste/Centro-Oeste

As tabelas 5,6,7 e 8 apresentam o resultado do erro quadrático médio normalizado na previsão do PLD, respectivamente para as técnicas ARIMA, rede neural recorrente, rede neural direta sem seleção de lags e rede neural direta com seleção de lags:

Tabela 5 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Sudeste/Centro-Oeste

Pesado	Médio	Leve
0.0058	0.0060	0.0058

Tabela 6 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Sudeste/Centro-Oeste

Pesado	Médio	Leve
0.0050	0.0054	0.0055

Tabela 7 – Erro médio quadrático normalizado utilizando rede direta sem seleção de lags no submercado Sudeste/Centro-Oeste

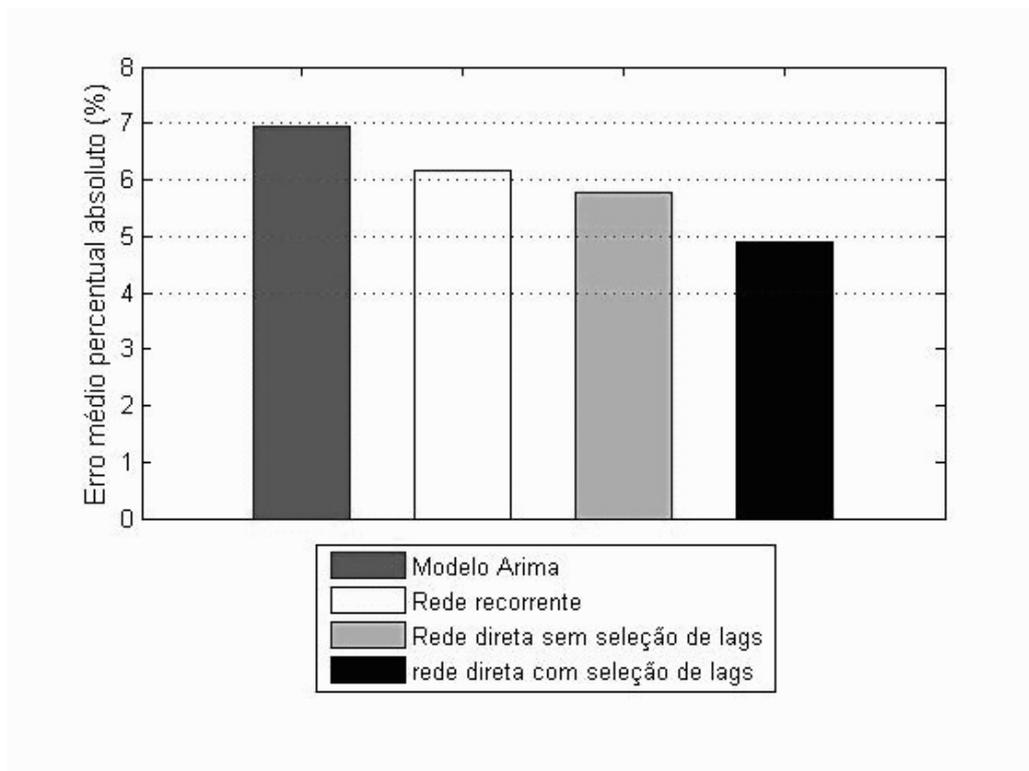
Patamar	Lags:1	Lags:2	Lags:3	Lags:4	Lags:5
Pesado	0.0058	0.0068	0.0063	0.0060	0.0055
Médio	0.0056	0.0061	0.0060	0.0056	0.0055
Leve	0.0059	0.0064	0.0061	0.0062	0.0050
	Lags:6	Lags:7	Lags:8	Lags:9	Lags:10
Pesado	0.0049	0.0079	0.0069	0.0069	0.0070
Médio	0.0052	0.0080	0.0074	0.0071	0.0073
Leve	0.0050	0.0072	0.0074	0.0061	0.0065

Tabela 8 – Erro quadrático médio normalizado utilizando rede neural direta com seleção de lags no submercado Sudeste/Centro-Oeste

limiar	Pesado	Médio	Leve
95%	0.0042	0.0045	0.0044
90%	0.0046	0.0049	0.0047

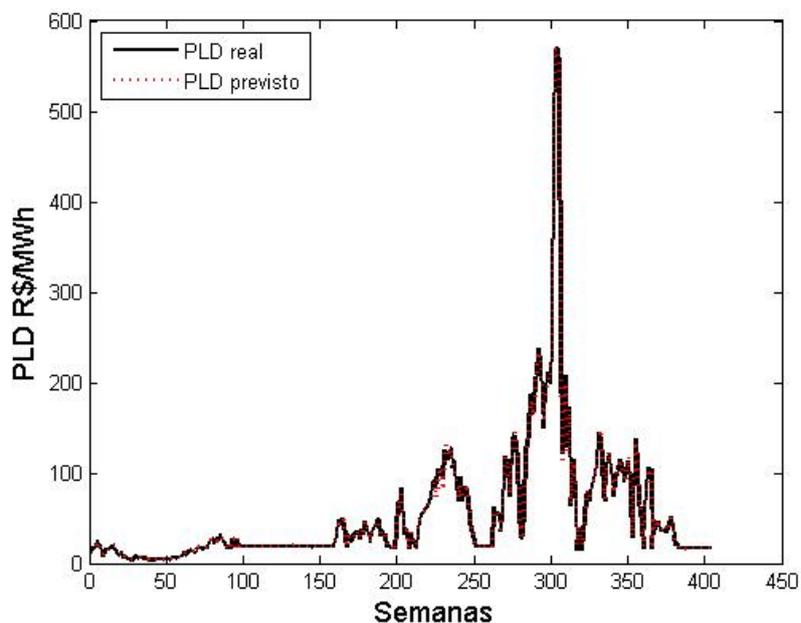
A figura 20 apresenta, em erro percentual absoluto, a média dos melhores desempenhos obtidos para cada técnica. Essa média é calculada sobre os erros dos patamares leve, médio e pesado. No caso da rede neural direta sem seleção de lags, a média será dos erros obtidos com 6 lags, pois foi o melhor resultado para essa técnica. E para o caso da rede neural direta com seleção de lags, a média calculada será a dos erros obtidos utilizando o limiar de 95% pelo mesmo motivo.

Figura 20 – Erro percentual médio absoluto no submercado Sudeste/Centro-Oeste



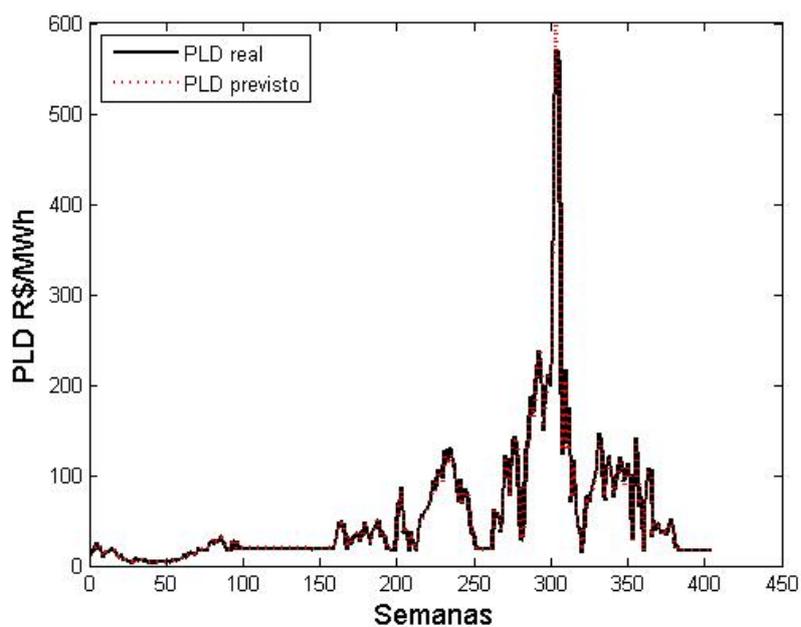
A figura 21 apresenta a previsão do PLD em patamar leve com as curvas do PLD real e o previsto pela rede direta com seleção de lags, a técnica que obteve o melhor desempenho no submercado Sudeste/Centro-Oeste:

Figura 21 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sudeste/Centro-Oeste com patamar de carga leve.



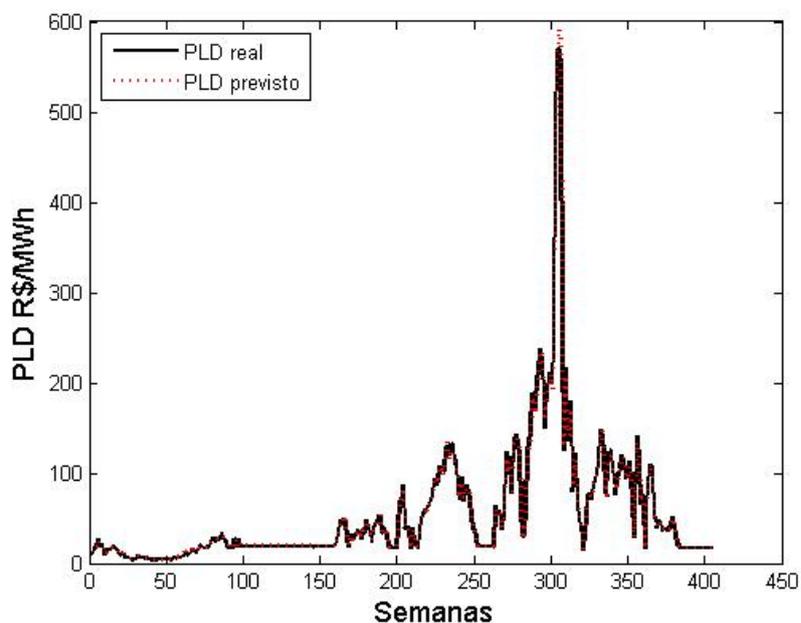
A figura 22 apresenta a previsão do PLD em patamar médio com as curvas do PLD real e o previsto pela rede direta com seleção de lags, a técnica que obteve o melhor desempenho no submercado Sudeste/Centro-Oeste:

Figura 22 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sudeste/Centro-Oeste com patamar de carga médio.



A figura 23 apresenta a previsão do PLD em patamar pesado com as curvas do PLD real e o previsto pela rede direta com seleção de lags, a técnica que obteve o melhor desempenho no submercado Sudeste/Centro-Oeste:

Figura 23 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sudeste/Centro-Oeste com patamar de carga pesado.



Os

5.2 Submercado Sul

As tabelas 9,10,11 e 12 apresentam o resultado do erro quadrático médio normalizado na previsão do PLD respectivamente para as técnicas ARIMA, rede neural recorrente, rede neural direta sem seleção de lags e rede neural direta com seleção de lags:

Tabela 9 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Sul

Pesado	Médio	Leve
0.0078	0.0075	0.0074

Tabela 10 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Sul

Pesado	Médio	Leve
0.0076	0.0073	0.0075

Tabela 11 – Erro quadrático médio normalizado utilizando rede direta sem seleção de lags no submercado Sul

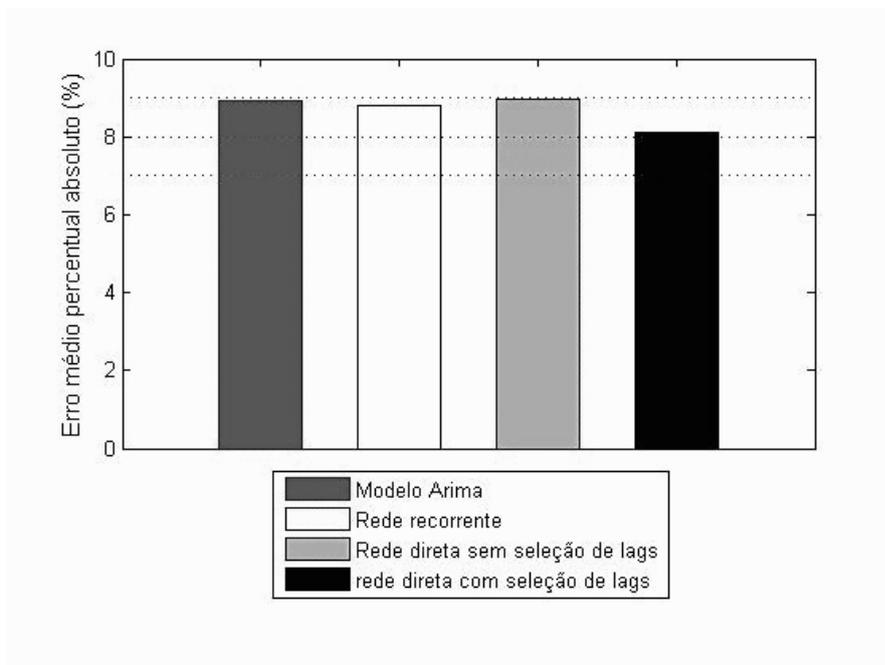
Patamar	Lags:1	Lags:2	Lags:3	Lags:4	Lags:5
Pesado	0.0079	0.0104	0.0136	0.0158	0.0163
Médio	0.0076	0.0097	0.0125	0.0129	0.0156
Leve	0.0073	0.0094	0.0125	0.0161	0.0173
	Lags:6	Lags:7	Lags:8	Lags:9	Lags:10
Pesado	0.0174	0.0233	0.0224	0.0227	0.0247
Médio	0.0179	0.0240	0.0224	0.0239	0.0229
Leve	0.0169	0.0228	0.0229	0.0224	0.0225

Tabela 12 – Erro quadrático médio normalizado utilizando rede direta com seleção de lags no submercado Sul

limiar	Pesado	Médio	Leve
95%	0.0070	0.0069	0.0067
90%	0.0075	0.0072	0.0068

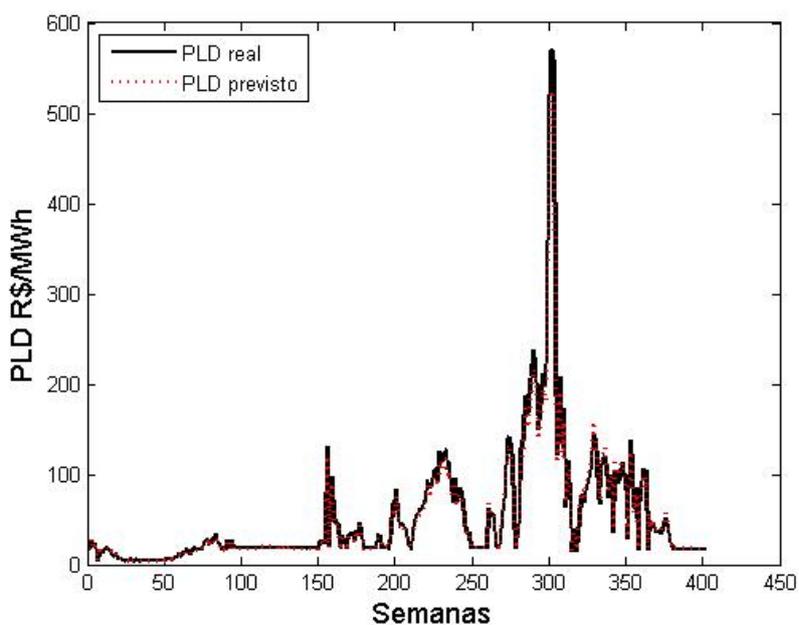
A figura 24 apresenta, em erro percentual absoluto, a média dos melhores desempenhos obtidos para cada técnica. Essa média é calculada sobre os erros dos patamares leve, médio e pesado. No caso da rede neural direta sem seleção de lags, a média será dos erros obtidos com 1 lag, pois foi o melhor resultado para essa técnica. E para o caso da rede neural direta com seleção de lags, a média calculada será a dos erros obtidos utilizando o limiar de 95%.

Figura 24 – Erro percentual médio absoluto no submercado Sul



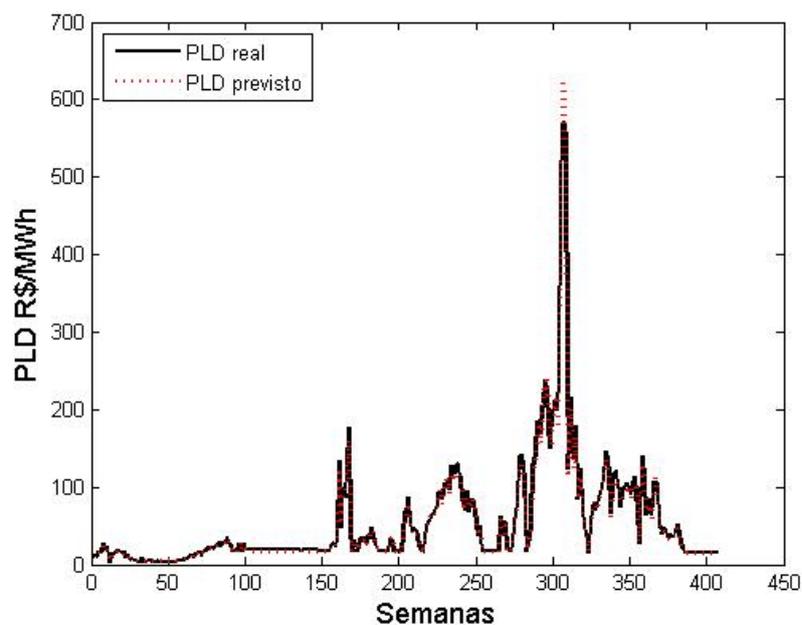
A figura 25 apresenta a previsão do PLD em patamar leve com as curvas do PLD real e o previsto pela rede direta com seleção de lags, a técnica que obteve o melhor desempenho no submercado Sul:

Figura 25 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sul com patamar de carga leve.



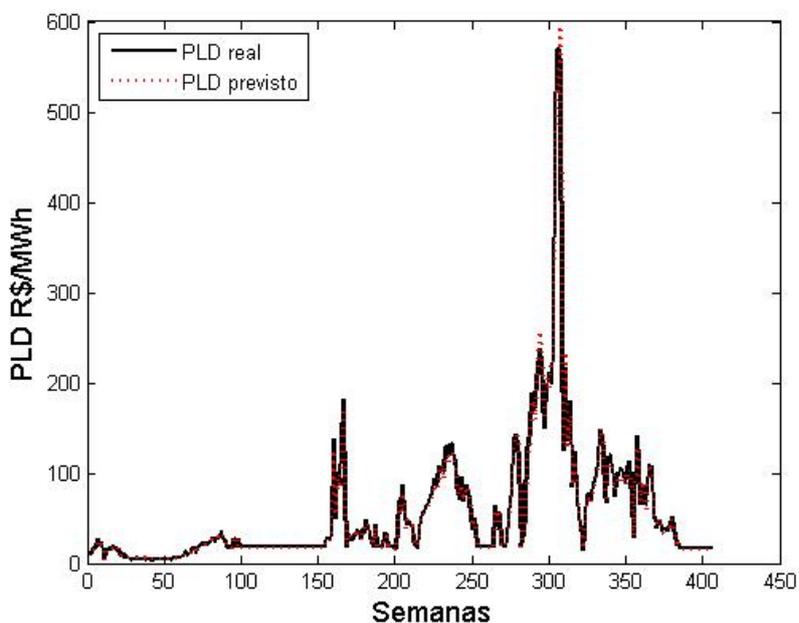
A figura 26 apresenta a previsão do PLD em patamar médio com as curvas do PLD real e o previsto pela rede direta com seleção de lags, a técnica que obteve o melhor desempenho no submercado Sul:

Figura 26 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sul com patamar de carga médio.



A figura 27 apresenta a previsão do PLD em patamar pesado com as curvas do PLD real e o previsto pela rede direta com seleção de lags, a técnica que obteve o melhor desempenho no submercado Sul:

Figura 27 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Sul com patamar de carga pesado.



5.3 Submercado Nordeste

As tabelas 13,14,15 e 16 apresentam o resultado do erro quadrático médio normalizado na previsão do PLD respectivamente para as técnicas ARIMA, rede neural recorrente, rede neural direta sem seleção de lags e rede neural direta com seleção de lags:

Tabela 13 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Nordeste

Pesado	Médio	Leve
0.0046	0.0046	0.0046

Tabela 14 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Nordeste

Pesado	Médio	Leve
0.0042	0.0042	0.0040

Tabela 15 – Erro quadrático médio normalizado utilizando rede direta sem seleção de lags no submercado Nordeste

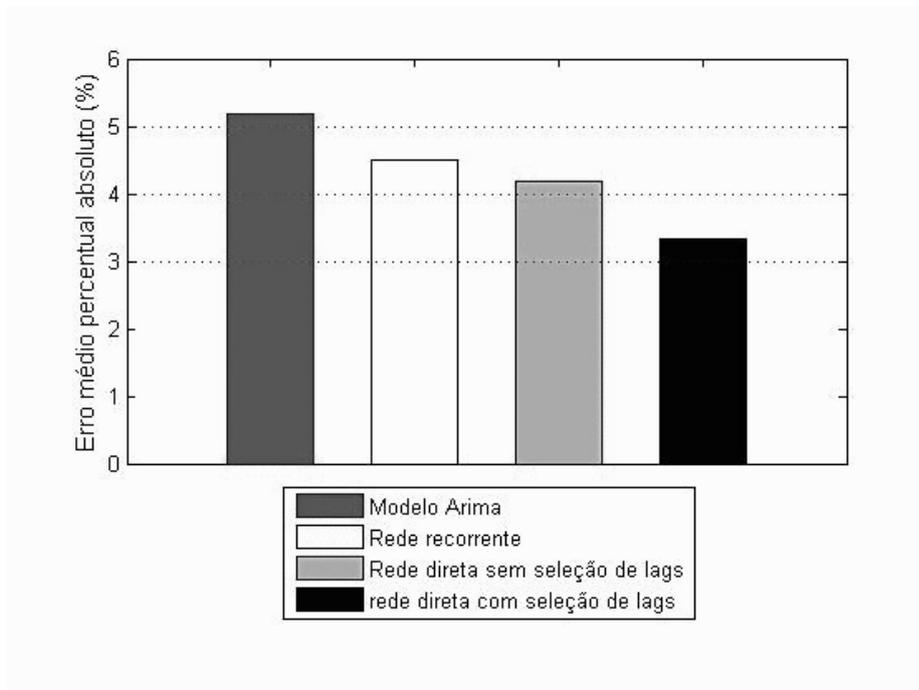
Patamar	Lags:1	Lags:2	Lags:3	Lags:4	Lags:5
Pesado	0.0042	0.0046	0.0045	0.0043	0.0041
Médio	0.0043	0.0044	0.0043	0.0042	0.0040
Leve	0.0042	0.0044	0.0042	0.0040	0.0039
	Lags:6	Lags:7	Lags:8	Lags:9	Lags:10
Pesado	0.0040	0.0059	0.0060	0.0062	0.0060
Médio	0.0040	0.0060	0.0062	0.0060	0.0064
Leve	0.0038	0.0059	0.0056	0.0064	0.0065

Tabela 16 – Erro quadrático médio normalizado utilizando rede direta com seleção de lags no submercado Nordeste

limiar	Pesado	Médio	Leve
95%	0.0035	0.0034	0.0033
90%	0.0038	0.0038	0.0037

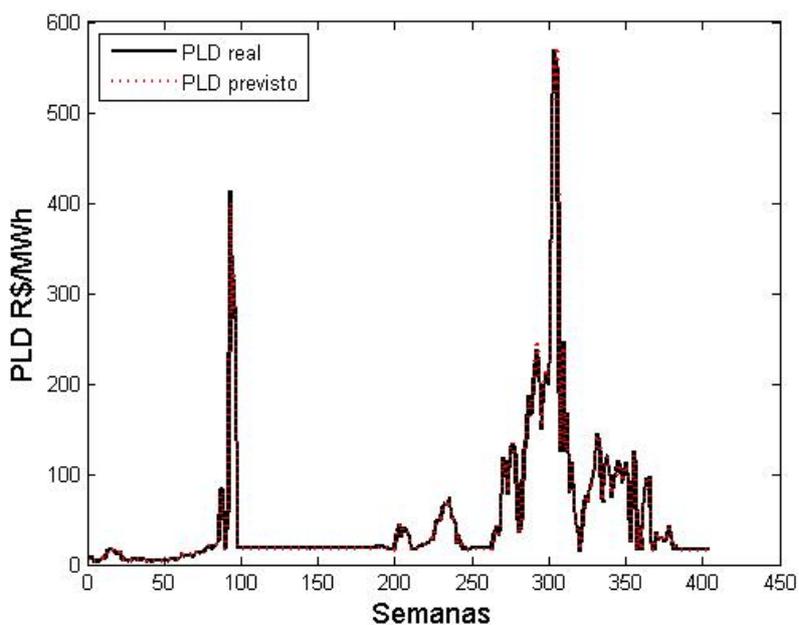
A figura 28 apresenta, em erro percentual absoluto, a média dos melhores desempenhos obtidos para cada técnica. Essa média é calculada sobre os erros dos patamares leve, médio e pesado. No caso da rede neural direta sem seleção de lags, a média é dos erros obtidos com 6 lags, pois foi o melhor resultado para essa técnica. E para o caso da rede neural direta com seleção de lags, a média calculada será a dos erros obtidos utilizando o limiar de 95%.

Figura 28 – Erro percentual médio absoluto no submercado Nordeste



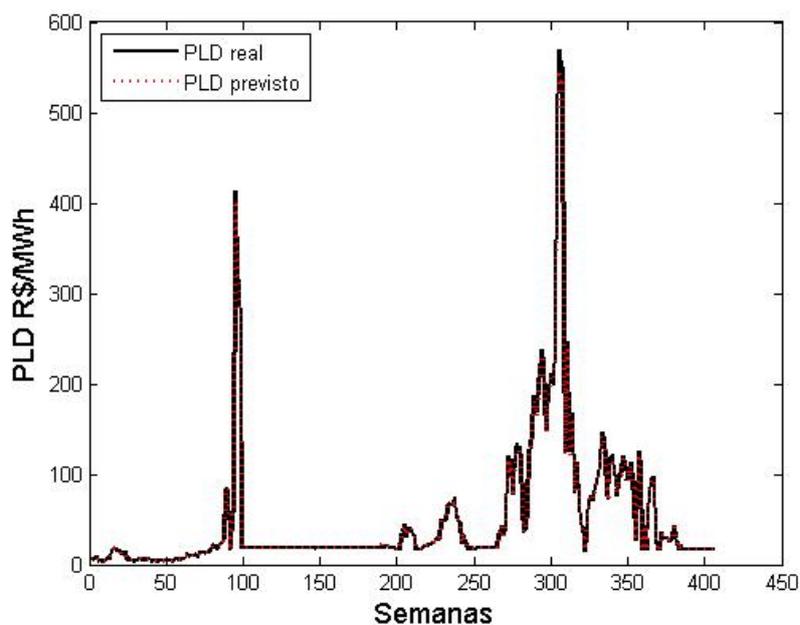
A figura 29 apresenta a previsão do PLD em patamar leve com as curvas do PLD real e o previsto pela rede direta com seleção lags, a técnica que obteve o melhor desempenho no submercado Nordeste:

Figura 29 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Nordeste com patamar de carga leve.



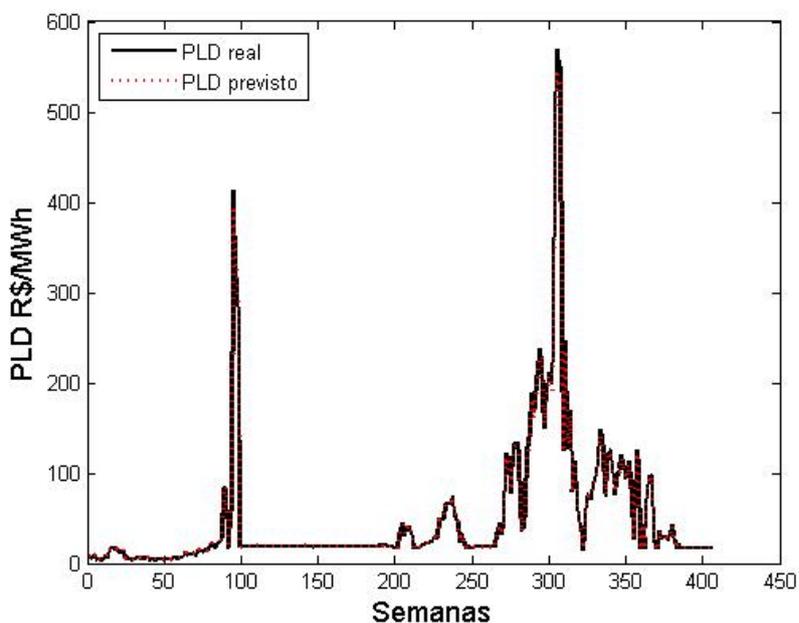
A figura 30 apresenta a previsão do PLD em patamar médio com as curvas do PLD real e o previsto pela rede direta com seleção lags, a técnica que obteve o melhor desempenho no submercado Nordeste:

Figura 30 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Nordeste com patamar de carga médio.



A figura 31 apresenta a previsão do PLD em patamar pesado com as curvas do PLD real e o previsto pela rede direta com seleção lags, a técnica que obteve o melhor desempenho no submercado Nordeste:

Figura 31 – PLD real x PLD previsto pela rede direta com seleção de lags. Submercado Nordeste com patamar de carga pesado.



5.4 Submercado Norte

As tabelas 17,18,19 e 20 apresentam o resultado do erro quadrático médio normalizado na previsão do PLD respectivamente para as técnicas ARIMA, rede neural recorrente, rede neural direta sem seleção de lags e rede neural direta com seleção de lags:

Tabela 17 – Erro quadrático médio normalizado utilizando modelo ARIMA no submercado Norte

Pesado	Médio	Leve
0.0090	0.0086	0.0088

Tabela 18 – Erro quadrático médio normalizado utilizando rede recorrente no submercado Norte

Pesado	Médio	Leve
0.0083	0.0081	0.0083

Tabela 19 – Erro quadrático médio normalizado utilizando rede direta sem seleção de lags no submercado Norte

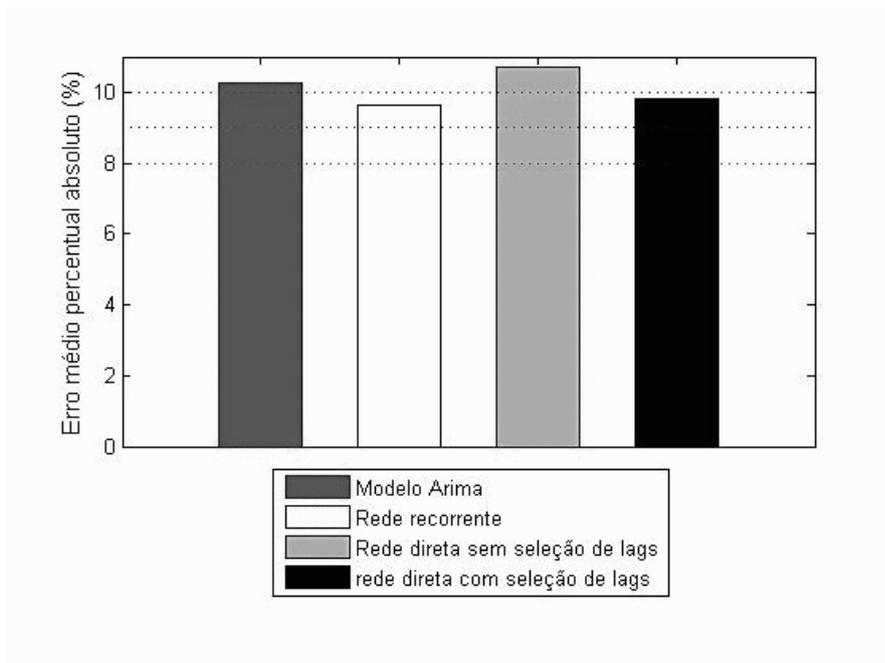
Patamar	Lags:1	Lags:2	Lags:3	Lags:4	Lags:5
Pesado	0.0108	0.0101	0.0098	0.0093	0.0102
Médio	0.0110	0.0098	0.0089	0.0089	0.0100
Leve	0.0122	0.0104	0.0090	0.0095	0.0097
	Lags:6	Lags:7	Lags:8	Lags:9	Lags:10
Pesado	0.0103	0.0104	0.0108	0.0111	0.0118
Médio	0.0104	0.0103	0.0109	0.0110	0.0117
Leve	0.0099	0.0105	0.0100	0.0104	0.0116

Tabela 20 – Erro quadrático médio normalizado utilizando rede direta com seleção de lags no submercado Norte

limiar	Pesado	Médio	Leve
95%	0.0088	0.0082	0.0082
90%	0.0091	0.0085	0.0086

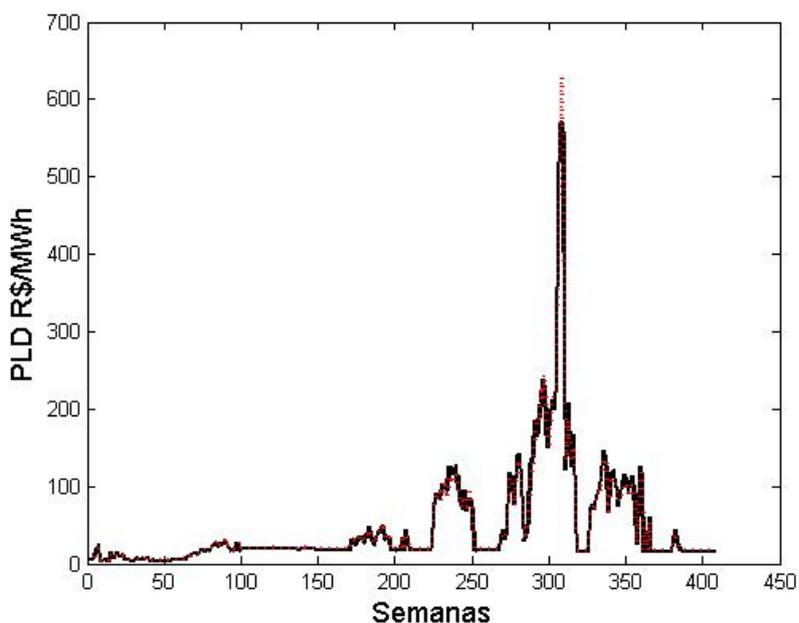
A figura 32 apresenta, em erro percentual absoluto, a média dos melhores desempenhos obtidos para cada técnica. Essa média é calculada sobre os erros dos patamares leve, médio e pesado. No caso da rede neural direta sem seleção de lags, a média será dos erros obtidos com 4 lags, pois foi o melhor resultado para essa técnica. E para o caso da rede neural direta com seleção de lags, a média calculada será a dos erros obtidos utilizando o limiar de 95%.

Figura 32 – Erro percentual médio absoluto no submercado Norte



A figura 33 apresenta a previsão do PLD em patamar leve com as curvas do PLD real e o previsto pela rede recorrente, a técnica que obteve o melhor desempenho no submercado Norte:

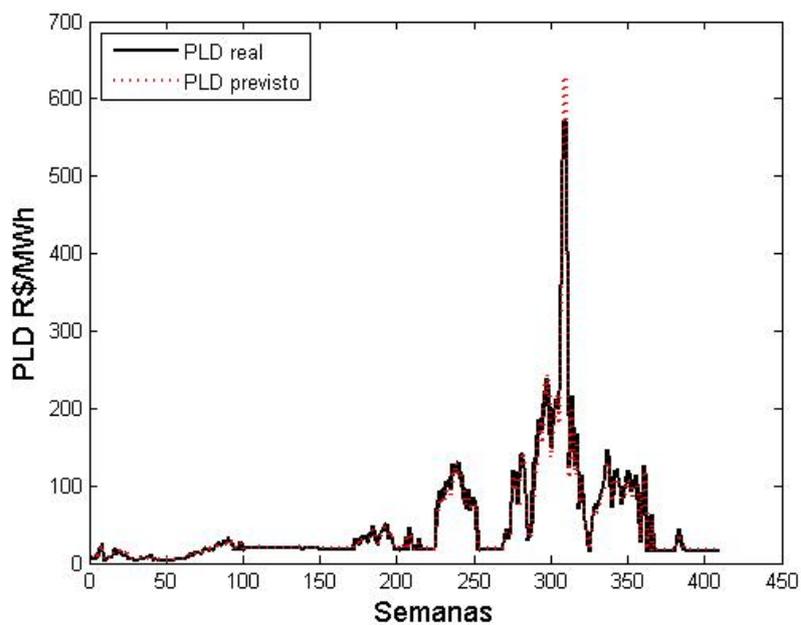
Figura 33 – PLD real x PLD previsto pela rede recorrente. Submercado Norte com patamar de carga leve.



A figura 34 apresenta a previsão do PLD em patamar médio com as curvas do PLD real e o previsto pela rede recorrente, a técnica que obteve o melhor desempenho no

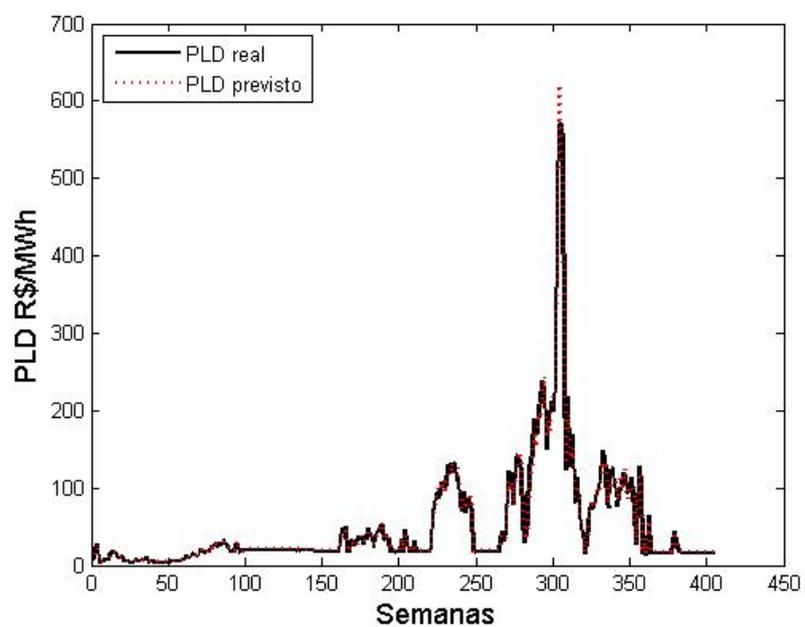
submercado Norte:

Figura 34 – PLD real x PLD previsto pela rede recorrente. Submercado Norte com patamar de carga médio.



A figura 35 apresenta a previsão do PLD em patamar pesado com as curvas do PLD real e o previsto pela rede recorrente, a técnica que obteve o melhor desempenho no submercado Norte:

Figura 35 – PLD real x PLD previsto pela rede recorrente. Submercado Norte com patamar de carga pesado.



O próximo capítulo apresenta as conclusões sobre as simulações realizadas referentes aos submercados, às técnicas utilizadas e o desempenho das mesmas.

6 CONCLUSÕES

Apresentam-se agora as conclusões. Primeiramente o comportamento das técnicas em cada submercado, após isso, uma análise do desempenho das redes neurais versus o desempenho do modelo ARIMA. Depois a análise passa para o desempenho da rede direta versus o desempenho da rede recorrente. É analisada também a contribuição do algoritmo de seleção de lags para o desempenho da rede neural direta. A relevância do estudo e a sua aplicação prática são reafirmadas e, finalmente, é apresentada uma seção de trabalhos futuros.

6.1 Submercados

Pode-se perceber que as variáveis de entrada comportam-se de maneira diferente em cada submercado, apresentando valores de correlação distintos. Essa diferença refletiu-se nos resultados de erro apresentados. No submercado Nordeste, as técnicas conseguiram um desempenho melhor comparado ao desempenho das mesmas nos outros submercados. Esse resultado era de certa forma esperado pois no capítulo 4 viu-se ver que as variáveis de entrada apresentaram maior valor de correlação com a saída nesse submercado. De forma análoga, o submercado Norte apresentou os maiores índices de erro, fato esperado já que nesse submercado a correlação das variáveis de entrada com a saída foi baixa comparada aos outros submercados.

6.2 Desempenho das redes neurais versus modelo ARIMA

Na literatura de previsão séries temporais é mostrado que o modelo ARIMA apresenta desempenho superior ao das redes neurais tradicionais treinadas com o algoritmo *back-propagation*, Reston, Affonso e Limão (2014) por exemplo. O melhor desempenho do modelo ARIMA acontece na grande maioria das séries reais (séries temporais com dados reais).

Neste trabalho pode-se mostrar como as melhorias implementadas (algoritmo Rprop, algoritmo de parada antecipada, algoritmo de seleção de lags para rede direta e a rede recorrente em si) elevaram o patamar das redes neurais de forma que as duas redes apresentaram desempenho superior ao do modelo ARIMA em todos os submercados, excetuando-se a rede neural sem seleção de lags que apresentou desempenho inferior ao modelo ARIMA no submercado Norte.

6.3 Desempenho da rede neural direta x rede neural recorrente

Em geral, a rede neural direta apresentou melhor desempenho que a rede recorrente obtendo erros inferiores em 3 submercados (Sudeste/Centro-Oeste, Sul e Nordeste). Esse desempenho sinaliza que a maior simplicidade da rede direta a tornou mais eficiente para essa série temporal particular nos submercados que apresentaram maior valor de correlação entrada-saída. No entanto, a rede recorrente superou o desempenho da rede direta no submercado Norte, que apresentou os menores valores de correlação entrada-saída. A maior complexidade e robustez da rede recorrente se mostrou como fator mais importante nesse submercado.

A título de informação adicional, pode-se citar que fatores como influência de ruído e presença de forte estado interno na série temporal favorecem o desempenho da rede neural recorrente em relação à rede direta. A rede recorrente consegue filtrar melhor e se comporta bem em situações 'adversas' se comparada à rede neural direta. Entretanto a série temporal analisada neste trabalho não apresentou essas características de forma significativa. Essa informação adicional pode ser vista em Connor, Martin e Atlas (1994) e Williams (1990).

6.4 Desempenho da rede neural direta sem seleção de lags x com seleção de lags

Pode-se comprovar a melhoria no desempenho da rede neural direta em todos os submercados ao se utilizar o algoritmo de seleção de lags. Como foi mostrado no capítulo 4, nem sempre a informação mais relevante para prever valores futuros de PLD é a informação mais recente.

Além disso mostrou-se que o limiar de 95% no algoritmo de seleção de lags foi mais eficiente por selecionar lags mais significativos em comparação ao limiar de 90%, que representa um relaxamento na condição implicando na seleção de mais lags menos significativos. O fato mostrado foi que a adição de mais informação com menos relevância não melhorou o desempenho da rede. O aumento de parâmetros acabou por degradar o desempenho mais que a adição de informação foi capaz de melhorá-lo.

6.5 Relevância do estudo e aplicação prática

É importante ressaltar que o trabalho realizado tem grande aplicação prática. A capacidade de prever com boa margem de erro o preço de energia elétrica de curto prazo no mercado brasileiro tem potencial uso comercial, empresas da região poderiam usar essa capacidade e possivelmente efetuar melhores decisões de mercado gerando economia. O mercado de energia brasileiro é singular, a forma de analisá-lo é distinta dos outros mercados como o europeu e o americano. O fato de existirem poucos trabalhos direcionados ao mercado brasileiro, torna este trabalho bastante relevante para a área de pesquisa de previsão de energia no mercado nacional.

6.6 Trabalhos futuros

Como trabalhos futuros, diversas melhorias e variações referentes à aplicação tratada neste trabalho podem ser implementadas. Com relação ao algoritmo de seleção de lags, é possível desenvolver a automatização da determinação do limiar, ou seja, determinar o limiar ótimo por meio de técnica específica. Além disso, a literatura está sempre aprimorando os algoritmos de treinamento para as redes neurais, logo, outras possibilidades de algoritmo podem ser testadas tanto para a rede recorrente quanto para a rede direta. Finalmente, é interessante também implementar a previsão do PLD para mais de um passo à frente, ampliando assim o horizonte de tempo para além de uma semana.

Referências

- ANBAZHAGAN, S.; KUMARAPPAN, N. Day-Ahead Deregulated Electricity Market Price Forecasting Using Recurrent Neural Network. *IEEE SYSTEMS JOURNAL*, v. 7, n. 4, Dezembro 2013. Citado na página 17.
- ANBAZHAGAN, S.; KUMARAPPAN, N. Day-ahead deregulated electricity market price forecasting using neural network input featured by DCT. *Energy Conversion and Management*, 2014. Citado na página 17.
- BRAGA, A. de P.; CARVALHO, A. P. de Leon E de; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. Rio de Janeiro: TC, 2000. Citado na página 30.
- CASTRO, N. J. de; LEITE, A. L. da S. Preço spot de eletricidade: teoria e evidências do caso brasileiro. 2008. Citado na página 15.
- CHAÂBANE, N. A hybrid ARFIMA and neural network model for electricity price prediction. *Electrical Power and Energy Systems*, 2014. Citado na página 17.
- CHOGUMAIRA, E. N.; HIYAMA, T. Short-Term Electricity Price Forecasting Using a Combination of Neural Networks and Fuzzy Inference. *Energy and Power Engineering*, Fevereiro 2011. Citado na página 18.
- C.IGEL; HUSK, M. Empirical evaluation of the improved Rprop learning algo- Rithm. *Neurocomputing*, 2003. Citado 3 vezes nas páginas 47, 49 e 59.
- CONNOR, J.; MARTIN, R.; ATLAS, L. E. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 1994. Citado na página 80.
- GHODSI, R.; ZAKERINIA, M. Forecasting Short Term Electricity Price Using Artificial Neural Network and Fuzzy Regression. *International Journal of Academic Research in Business and Social Sciences*, v. 2, n. 1, 2012. ISSN 2222-6990. Citado na página 18.
- GUPTA, A.; CHAWLA, P.; CHAWLA, S. Short Term Electricity Price Forecasting Using ANN and Fuzzy Logic under Deregulated Environment. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, v. 2, Agosto 2013. Citado na página 18.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2. ed. [S.l.]: Prentice Hall, 1999. ISBN 81-7808-300-0. Citado 3 vezes nas páginas 30, 38 e 47.
- HONG, Y.; WU, C. Day-Ahead Electricity Price Forecasting Using a Hybrid Principal Component Analysis Network. *Energies*, Novembro 2012. ISSN 1996-1073. Citado na página 17.
- KEYNIA, F. A new feature selection algorithm and composite neural network for electricity price forecasting. *Engineering Applications of Artificial Intelligence*, Fevereiro 2012. Citado na página 17.
- LUKOŠEVIČIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, v. 3, Agosto 2009. Citado na página 34.

- MAYO, R. *Derivativos de eletricidade & gerenciamento de risco*. Rio de Janeiro: Synergia, 2009. Citado 2 vezes nas páginas 15 e 24.
- PHAM, D.; LIU, X. Training of Elman networks and dynamic system modeling. *International Journal of Systems Science*, 1996. Citado na página 36.
- PRECHELT, L. Automatic early stopping using cross validation: Quantifying criteria. *Neural networks*, 1998. Citado 2 vezes nas páginas 51 e 52.
- RESTON, J. C.; AFFONSO, C. de M.; LIMÃO, R. C. Energy price prediction multi-step ahead using hybrid model in the Brazilian market. *Electric Power Systems Research*, Agosto 2014. Citado 2 vezes nas páginas 18 e 79.
- RIEDMILLER, M. Rprop - Description and implementation details. *Technical Report*, 1994. Citado na página 47.
- SHARMA, V.; D.SRINIVASAN. A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market. *Engineering Applications of Artificial Intelligence*, Março 2013. Citado na página 17.
- SILVA, E. L. da. *Formação de preços em mercados de energia elétrica*. Porto Alegre: Sagra Luzzato, 2001. Citado na página 25.
- SOUZA, F. C. de. *Dinâmica da gestão de riscos no ambiente de contratação regulada do setor elétrico brasileiro*. 2008. Tese (Doutorado) — Universidade Federal do Rio de Janeiro. Citado na página 20.
- VEELENBURF, L. P. J. *Analysis and Applications of Artificial Neural Networks*. 1. ed. [S.l.]: Prentice Hall, 1995. Citado na página 31.
- WAN, C. et al. A hybrid approach for probabilistic forecasting of electricity price. *IEEE TRANSACTIONS ON SMART GRID*, v. 5, n. 1, Janeiro 2014. Citado na página 17.
- WILLIAMS, R. J. Adaptive state representation and estimation using recurrent connectionist networks. In: _____. *Neural networks for control*. [S.l.]: MIT Press, 1990. cap. 4, p. 97 – 114. ISBN 0-262-13261-3. Citado 3 vezes nas páginas 34, 36 e 80.