UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Aplicação de Modelos Neurais na Previsão de Séries Temporais

João Paulo Reus Rodrigues Leite

Itajubá, dezembro de 2010

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

João Paulo Reus Rodrigues Leite

Aplicação de Modelos Neurais na Previsão de Séries Temporais

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Otávio Augusto Salgado Carpinteiro

Dezembro de 2010
Itajubá - MG

Ficha catalográfica elaborada pela Biblioteca Mauá –
Bibliotecária Margareth Ribeiro- CRB_6/1700

L533a

Leite, João Paulo Reus Rodrigues
Aplicação de modelos neurais na previsão de séries temporais / João Paulo Reus Rodrigues Leite. -- Itajubá, (MG) : [s.n.], 2010.
104 p. : il.

Orientador: Prof. Dr. Otávio Augusto Salgado Carpinteiro.
Dissertação (Mestrado) – Universidade Federal de Itajubá.

1. Redes neurais. 2. Mapa auto-organizável. 3. Máquina de vetor de suporte. 4. Modelo hierárquico. 5. Previsão em séries temporais financeiras. 6. Função de ações. I. Carpinteiro, Otávio Augusto Salgado, orient. II. Universidade Federal de Itajubá. III. Título.

Agradecimentos

Agradeço a Deus pela minha vida e minha saúde. Que eu possa retribuir a oportunidade de ter nascido cercado de pessoas tão boas e de viver em um lugar que gosto tanto utilizando da melhor maneira possível os dons a mim concedidos por Ele.

Aos meus pais, Carlos e Maria, por serem exemplos de vida e fontes de amor incondicional. Seu apoio, compreensão e otimismo constantes foram fundamentais e sempre me motivaram acima de qualquer outro fator.

Ao Prof. Dr. Otávio Carpinteiro, por fazer valer a posição de orientador, guiando decisivamente os passos da pesquisa. Agradeço por ter me ajudado a explorar todo meu potencial e nunca ter deixado de incentivar a busca pelo melhor resultado, mesmo nos momentos menos favoráveis. Agradeço também aos professores e colegas pesquisadores do Grupo de Pesquisas em Engenharia de Sistemas e de Computação (GPESC), que se mostraram sempre disponíveis e contribuíram inúmeras vezes para o prosseguimento do trabalho, através de sugestões, conselhos e discussões.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio financeiro concedido durante dois anos da pesquisa e aos amigos da Core Networks Ltda., que me acolheram após o término da bolsa e sempre foram flexíveis em relação aos meus compromissos com o mestrado.

Finalmente, gostaria de agradecer a todas as pessoas que estiveram comigo, mesmo que brevemente, no período em que desenvolvi a pesquisa. Nenhuma pessoa é feita apenas de si mesma, mas formada por milhares de outras. Cada pessoa que alguma vez tenha me acompanhado, inspirado, ou dito uma palavra de encorajamento, entrou na construção dos meus pensamentos, e compartilha também do meu sucesso. Quando acendemos uma tocha para iluminar o caminho de outra pessoa, clareamos também o nosso próprio caminho.

A felicidade só é real quando compartilhada.

Chris McCandless, Na Natureza Selvagem

Resumo

Este estudo apresenta uma abordagem original para previsões de valores em séries temporais. O objetivo deste trabalho é o desenvolvimento e validação de um novo modelo baseado em uma arquitetura hierárquica, sua aplicação em uma série histórica real de fundo de investimentos e, por fim, a comparação de seu desempenho com o de outras arquiteturas mais tradicionais, como o MLP e a SVM. Através de sua estrutura, composta por um mapa auto-organizável (SOM) e uma máquina de vetor de suporte (SVM), deseja-se processar os dados do espaço de entrada, extraindo suas características estatísticas mais importantes e inserindo-os em um contexto, esperando, com isso, alcançar um desempenho de previsão superior aos modelos tradicionais.

Dados de séries de fundo de investimentos se apresentam geralmente em agrupamentos bem separados, ou *clusters*, que se revezam ciclicamente no tempo e se caracterizam por comportamentos distintos, onde a série demonstra maior ou menor volatilidade. Por este motivo, em uma segunda etapa do estudo, uma nova arquitetura foi desenvolvida, composta por dois modelos hierárquicos, especializados em comportamentos distintos, e uma fase inicial, responsável pela segmentação da série em períodos de alta ou baixa volatilidade.

Ambas as arquiteturas se mostraram superiores aos modelos estabelecidos como parâmetros de comparação, em relação tanto ao erro percentual absoluto médio obtido nos experimentos quanto, também, à captura da dinâmica da série, comprovando que a manipulação eficiente de informações de contexto gera benefícios para o modelo e resulta em previsões de maior qualidade.

Abstract

This study presents an original approach to time series forecasting. Its objective is the development and validation of a new model based on a hierarchical architecture, its application in a real series of stock market fund and, finally, the comparison of its performance with those of more traditional architectures, such as MLP and SVM. Through its structure, made up by a self-organizing map (SOM) and a support vector machine (SVM), we want to process the data of the input space, extracting its most important statistical characteristics and inserting them into a context, in order to achieve a performance superior to that of traditional forecast models.

Series data of stock market fund usually present themselves in well-separated clusters, which alternate cyclically in time and are characterized by distinct behaviors, where the series shows a higher or lower volatility. For this reason, in a second stage of the study, a new architecture was developed, consisting of two hierarchical models, specialized on different behaviors, and an initial phase, responsible for the segmentation of the series on periods of high or low volatility.

Both architectures were superior to the models used as comparison parameters, for both the mean absolute percentual error obtained in the experiments and the capture of the dynamics of the series, proving that efficient handling of context information generates benefits to forecast models and results in high quality predictions.

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

ARCH	*Auto-Regressive Conditional Heteroscedastic*
ARIMA	*Auto-Regressive Integrated Moving Average*
BOVESPA	Bolsa de Valores de São Paulo
C-3PO	*Class-3 Protocol Officer*
CBR	*Case-Based Reasoning*
HNM	Modelo Neural Hierárquico
HNM-V	Modelo Neural Hierárquico com Volatilidade
IBrX	Índice Brasil
KM	*Kernel Methods*
LIBSVM	Biblioteca de software para Máquinas de Vetor de Suporte
MAE	*Mean Absolute Error*
MAPE	*Mean Absolute Percentage Error*
ME	*Mean Error*
MLP	*Multi-Layer Perceptron*
MPE	*Mean Percentage Error*
MSE	*Mean Squared Error*
RBF	*Radial-Basis Function*
SLP	*Single Layer Perceptron*
SOM	*Self-Organizing Map*
SVC	*Support Vector Classification*
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*

1 Introdução

1.1 Descrição do Problema

O aprimoramento das técnicas de previsão em séries temporais é um assunto de muita importância. A simples possibilidade da previsão confiável do comportamento de uma determinada variável de interesse pode se tornar um fator fundamental no processo de tomada de decisão. Meteorologistas, cientistas, analistas econômicos e, é claro, investidores que detêm conhecimento do futuro estão literalmente um passo a frente de seus concorrentes, mesmo que a previsão consista apenas em um forte indicativo ou tendência.

O objetivo deste trabalho é o desenvolvimento e validação de um novo modelo para previsão em séries temporais financeiras, baseado em uma arquitetura hierárquica de redes neurais bem-sucedida em problemas de outras naturezas (CARPINTEIRO et al., 2007), sua aplicação em uma série histórica real de fundo de investimentos e, por fim, a comparação de seu desempenho com outras arquiteturas mais tradicionais, como o perceptron de múltiplas camadas (MLP) e a máquina de vetor de suporte (SVM). Através de sua estrutura, composta por um mapa auto-organizável (SOM) e uma máquina de vetor de suporte (SVM), deseja-se realizar um processamento mais refinado nos dados do espaço de entrada, através da extração de suas características mais importantes e sua inserção em um contexto histórico e estatístico, esperando, com isso, alcançar um desempenho de previsão superior aos modelos tradicionais.

A análise de séries temporais financeiras preocupa-se basicamente com a teoria e prática da avaliação de um ativo sobre o tempo (TSAY, 2002), e proporciona ao analista um grupo de ferramentas que auxiliam na definição de estratégias. O que torna a previsão ainda mais complicada são duas características inerentes a este tipo de série temporal: Ruído e não-estacionariedade. Segundo Cao (2002), o ruído se refere à indisponibilidade da informação completa relacionada ao comportamento passado da série, de maneira que se torna difícil a captura da dependência existente entre o futuro e o passado. Já a característica de não-estacionariedade implica que a dinâmica da série muda em diferentes

regiões ou períodos de tempo, levando a uma mudança gradual e sensível na dependência entre a entrada e a saída do modelo de previsão.

Para explorar todo o conteúdo de informação destes sinais, necessitamos de máquinas que processem algoritmos baseados em inteligência artificial, cujo projeto inclua ainda questões fundamentais como a não-linearidade, responsável pela extração das características específicas dos sinais de entrada, e capacidade de aprendizagem e adaptação, através da qual o modelo se torna capaz de aprender a dinâmica da série e se adaptar às mudanças estatísticas lentas do ambiente, de maneira contínua.

As redes neurais representam um tema multidisciplinar, com raízes na neurociência, matemática, estatística, física, ciência da computação e engenharia. Sua habilidade para aprender a partir de dados dotou-as com uma propriedade poderosa, de implicações tanto teóricas quanto práticas. De uma forma ou de outra, a habilidade das redes neurais de aprender a partir de exemplos representativos de seu ambiente as torna ferramentas úteis em aplicações tão diversas quanto modelagem, reconhecimento de padrões, processamento de sinais, controle e análise de séries temporais.

Podemos dizer que as redes neurais têm muito a oferecer em vários campos de aplicação, especialmente quando a solução de um problema de interesse é dificultada por alguns pontos, como a falta de entendimento físico ou estatístico do problema, variações estatísticas nos dados observáveis e, principalmente, quando um mecanismo não-linear é responsável pela geração dos dados. Seus algoritmos de aprendizagem geralmente eliminam a necessidade de extração manual de características do conjunto de dados, permitindo treinar extratores de características, classificadores, processadores contextuais e modelos de previsão, que são o objeto de estudo deste trabalho.

Portanto, as séries financeiras, de natureza não-estacionária e complexa, governadas por dinâmicas não-lineares, representam um desafio para uma descrição matemática exata mas se apresentam como candidatas ideais para o tratamento com redes neurais artificiais.

1.2 Necessidade de uma Nova Abordagem

Redes neurais artificiais apresentam-se como uma alternativa mais atrativa para tarefas de previsão devido às suas características básicas. Primeiro, diferente dos modelos estatísticos, redes neurais são métodos auto-adaptativos baseados apenas nos dados. Elas aprendem a partir de exemplos e capturam os relacionamentos funcionais da série sem a necessidade de conhecimento prévio sobre o relacionamento entre as variáveis de entrada

e saída, mostrando-se mais gerais e flexíveis. Além disso, redes neurais são aproximadores de função universais e podem generalizar a partir de seu conhecimento, inferindo a parte desconhecida dos dados mesmo que os exemplos apresentados em sua entrada sejam ruidosos, desde que estes dados estejam dentro de um intervalo semelhante ao utilizado durante o treinamento.

As abordagens estatísticas tradicionais, como Box-Jenkins (BOX; JENKINS, 1976) ou o método ARIMA (PANKRATZ, 1983) assumem que a série sob estudo é gerada a partir de um processo linear. Estes modelos podem ser analisados em detalhe e possuem a vantagem de serem compreendidos mais facilmente, além de possuírem implementação computacional mais simples. No entanto, eles podem ser completamente inapropriados se o mecanismo que origina a série for não-linear e, na verdade, sistemas do mundo real são frequentemente não-lineares.

A utilização de modelos estatísticos não-lineares, como o ARCH, ou *auto-regressive conditional heteroscedastic* (ENGLE, 1982), também possui desvantagens, pois a previsão fica ainda limitada pela necessidade da descrição explícita dos relacionamentos existentes na série, realizada mesmo que haja pouco conhecimento sobre o processo que a origina. Na verdade, a formulação de um modelo não-linear para um conjunto de dados particular é uma tarefa de grande dificuldade, uma vez que existem muitos padrões de não-linearidade e um modelo pré-especificado está sujeito a não ser geral o bastante para capturar todas as características importantes do sistema analisado.

A busca por modelos que levem a resultados de previsão mais precisos tem sido responsável por avanços tanto na área de modelagem estatística quanto na de redes neurais. A pesquisa na área de redes neurais, no entanto, tem sido mais intensa, principalmente porque estes modelos têm demonstrado melhor desempenho (HILL; O'CONNOR; REMUS, 1996). Portanto, torna-se uma alternativa promissora o desenvolvimento de modelos de previsão baseados em redes neurais, que sejam cada vez mais precisos e específicos para cada tipo de série temporal.

Na previsão em séries temporais financeiras, segundo Cao (2002), o modelo mais popular continua sendo o perceptron de múltiplas camadas com algoritmo de retropropagação, devido à sua arquitetura simples e boa capacidade na solução de problemas. No entanto, este tipo de rede apresenta alguns pontos negativos, incluindo um número alto de parâmetros livres e dificuldades na obtenção de uma solução única e estável. Por este motivo, novas técnicas vêm ganhando mais espaço e entre elas destaca-se a máquina de vetor de suporte, cujo funcionamento leva o modelo sempre em direção a uma solução global e

ótima, utilizando-se de poucos parâmetros. Seus resultados já se mostraram superiores aos da MLP por diversas oportunidades (KIM, 2003) (CAO; TAY, 2001).

A inclusão de outros modelos neurais, como os mapas auto-organizáveis, na análise de séries temporais financeiras proporciona novas possibilidades na construção de previsores com estruturas hierárquicas. Nestes modelos, o papel da rede SOM é realizar a segmentação dos dados de entrada em subgrupos, enquanto um conjunto de redes neurais especializadas realiza a previsão para cada um destes subgrupos formados, a partir de um treinamento com aprendizagem supervisionada - geralmente máquinas de vetor de suporte (HSU et al., 2009) (TAY; CAO, 2001b). A nova abordagem, proposta nesta dissertação, também utiliza um modelo hierárquico composto por dois estágios: um mapa auto-organizável e uma máquina de vetor de suporte. A diferença entre o novo modelo e o modelo utilizado por outros pesquisadores reside na maneira como é utilizado o mapa auto-organizável.

Segundo Kohonen (1990), para descrever o papel de um item no comportamento da série, é necessário que esse dado seja apresentado com uma quantidade suficiente de contexto. Qualquer tarefa de processamento necessita da organização da informação disponível e, em consequência disso, muitos dos modelos que processam os valores de entrada sem estruturação exibem baixa convergência e pouca eficiência no processo de generalização. Neste novo modelo, o papel da rede SOM é construir um contexto para os dados de entrada, codificando-os em um mapa de características onde a informação se encontre de forma concentrada. Ao apresentar estes novos dados à entrada da máquina de vetor de suporte, espera-se obter um ganho considerável na sua capacidade de previsão, através de um maior conhecimento adquirido a partir da série financeira estudada.

1.3 Estrutura da Dissertação

Nos últimos anos, redes neurais artificiais têm sido utilizadas com sucesso na modelagem de séries temporais financeiras (YANG; CHAN; KING, 2002) (KIM, 2003). Através deste trabalho, foi realizado um esforço para a superação dos problemas característicos deste tipo de série através da escolha de uma arquitetura adequada de redes neurais para a prospecção de suas características fundamentais.

O segundo capítulo fornece a fundamentação teórica necessária para a compreensão do modelo desenvolvido e dos testes realizados. Nele, são explicados conceitos sobre as séries temporais, suas medidas estatísticas e de desempenho, e as características próprias

de séries financeiras. Na sequência, também são explorados os conceitos fundamentais de cada um dos tipos de redes neurais utilizados neste trabalho: Perceptron de multiplas camadas (MLP), mapas auto-organizáveis (SOM) e máquinas de vetor de suporte (SVM). O final do capítulo apresenta uma seção especial com a revisão sobre os trabalhos realizados neste campo de pesquisa.

O terceiro capítulo apresenta a proposta de um novo modelo para a previsão em séries financeiras, o Modelo Neural Hierárquico - HNM (do inglês *Hierarchical Neural Model*). Neste capítulo são esclarecidas as motivações para sua construção, seus componentes estruturais e o funcionamento detalhado de seu mecanismo de previsão.

O capítulo 4 contém todas as informações relacionadas aos experimentos realizados para a validação do modelo. Inicia-se com a apresentação da série utilizada no decorrer dos testes e segue a ordem cronológica dos acontecimentos da pesquisa, detalhando cada passo e cada descoberta e apresentando, ao final, uma análise quantitativa e qualitativa dos resultados alcançados.

O último capítulo conclui a dissertação, verificando se os objetivos iniciais foram alcançados e apresentando as considerações finais sobre o estudo, além de sugestões para a continuidade do trabalho.

2 Fundamentação Teórica

2.1 Previsão em Séries Temporais Financeiras

2.1.1 Séries Temporais

O conceito de série temporal é muito intuitivo e utilizado com muita frequência no cotidiano, mesmo que quase sempre de maneira informal. Segundo Makridakis, Wheelwright e Hyndman (1998), é chamada de série temporal "uma sequência ordenada de valores de uma variável observados em intervalos de tempo igualmente espaçados". Seguindo a mesma linha, Bowerman, O'Connell e Koehler (2005) nos dizem que uma série temporal é uma "sequência cronológica de observações de uma variável em particular". As definições possuem aspectos em comum que formam o núcleo da teoria de séries temporais: a observação do comportamento de uma variável no decorrer tempo.

Antes de iniciar o estudo das características inerentes a uma série temporal relacionada ao mercado financeiro é preciso conhecer bem os conceitos e as ferramentas disponíveis para o tratamento de séries temporais sob um aspecto geral. Alguns exemplos de séries temporais podem ser retirados das mais diversas áreas de pesquisa:

- Consumo mensal de energia elétrica de uma fábrica durante os últimos dez anos;
- Temperatura de uma reação química a cada segundo no intervalo de cinco minutos consecutivos;
- Valor diário do preço de cotas de um índice financeiro no período de cinco anos.

A análise de séries temporais pode ser realizada com diversas finalidades. Através dos dados disponíveis é possível produzir sumários para esses dados, construir histogramas, identificar a existência de componentes sazonais e tendências, estabelecer uma relação de causalidade entre a variável e seus valores passados, ou mesmo classificar os dados de acordo com algum critério. No entanto, uma das maiores contribuições do estudo de séries

temporais é sua análise para a descoberta de um padrão histórico de comportamento, que torna-se ferramenta fundamental no projeto de modelos de previsão.

O desenvolvimento de um previsor confiável pode se tornar um fator de grande importância em um processo de tomada de decisão. O conhecimento do futuro faz com que seu portador esteja sempre um passo a frente dos demais, possuindo vantagem no processo de definição de estratégias e no ataque a qualquer problema que se desenvolva no tempo. Neste sentido, faz-se conveniente o estabelecimento e a identificação de alguns padrões de comportamento recorrentes (BOWERMAN; O'CONNELL; KOEHLER, 2005), que podem aparecer sozinhos ou combinados, facilitando ou dificultando a modelagem de uma série temporal. São eles:

Horizontal: Este padrão ocorre quando os dados da série flutuam em volta de um valor constante, fazendo com que a série apresente uma média aproximadamente constante por toda sua extensão;

Tendência: Caracterizado por um aumento ou diminuição a longo prazo nos dados, gerando uma representação gráfica com aspecto de "rampa" de subida ou descida;

Sazonal: É um padrão de aspecto periódico, onde um fator sazonal influencia o comportamento da série, que se repete dentro de um período fixo de tempo;

Cíclico: Identificado por elevações e quedas cíclicas que ocorrem de tempos em tempos na série, mas sem período fixo de tempo, podendo, portanto, variar de duração;

Flutuações irregulares: São movimentos erráticos nos dados, causados muitas vezes por fatores externos e imprevisíveis. São frequentemente chamados de "*outliers*" ou "pontos fora da curva".

É importante lembrar que os comportamentos descritos acima nem sempre ocorrem sozinhos em uma série temporal e, geralmente, aparecem na forma de componentes que se combinam para formar o comportamento da série como um todo. Segundo Bowerman, O'Connell e Koehler (2005), esta é a razão pela qual não existe um modelo previsor universal. Um modelo destinado a prever séries com comportamento forte de tendência pode não funcionar adequadamente para uma série onde as componentes sazonal e horizontal são dominantes. Assim, é papel do desenvolvedor identificar as características mais evidentes na série estudada e desenvolver o modelo de previsão adequado para seu problema em específico.

2.1.2 Medidas Estatísticas

A análise das séries temporais se apóia firmemente nas bases da estatística. Muitos valores numéricos de natureza estatística são utilizados com a finalidade de descrever com precisão o comportamento da série no decorrer do tempo e identificar suas principais características. As seções seguintes (denotadas entre parênteses) têm como objetivo estabelecer o vocabulário e os fundamentos básicos para a análise da série que servirá como objeto de estudo desta dissertação, definindo medidas como média aritmética (2.1.2.1), desvio padrão (2.1.2.3), variância e covariância (2.1.2.2), além dos conceitos fundamentais de estacionariedade (2.1.2.4) e autocorrelação (2.1.2.5).

2.1.2.1 Média Aritmética

A média aritmética simples é amplamente utilizada no nosso dia-a-dia. A média de um conjunto de valores numéricos é calculada somando-se todos estes valores e dividindo-se o resultado pela quantidade de elementos somados. Seu resultado, representado pelo símbolo $\bar{x}$, é um sumário numérico que representa o centro de gravidade do conjunto de dados e serve como um valor de referência para este mesmo conjunto. Se tivermos uma série de n valores de uma variável x, a média aritmética simples será determinada pela equação (2.1):

$$\bar{x} = \frac{x_1 + x_2 + ... + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.1}$$

2.1.2.2 Variância e Covariância

O cálculo da média aritmética raramente é suficiente para caracterizar um conjunto de dados. Dados muito diferentes da média podem estar no conjunto e é necessário, portanto, introduzir uma medida de dispersão, que defina o quão longe estes valores se encontram da média, dispersos pelo espaço. Na teoria da probabilidade e na estatística, a variância de uma variável aleatória é uma medida da sua dispersão estatística, indicando quão longe, em geral, os seus valores se encontram do valor esperado. A variância é expressa pelo símbolo s^2 e definida pela equação (2.2):

$$s^2 = Var(x) = \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2 \tag{2.2}$$

Pode-se notar que o primeiro passo é o cálculo do desvio $x_i - \bar{x}$. A soma destes desvios, no entanto, será sempre igual a zero. Para que os dados de desvio sejam sempre positivos, utiliza-se o desvio elevado ao quadrado. A unidade do resultado final é, também, dada pela unidade original da observação ao quadrado. Este fato motivou os estatísticos a utilizarem como medida mais comum para a dispersão a raiz quadrada da variância, conhecida como desvio padrão s.

Outra medida cuja definição se faz necessária é a da covariância entre duas variáveis. Em muitos casos, é importante demonstrar numericamente o relacionamento de dependência entre duas variáveis distintas, de maneira a examinar o quanto uma mudança em uma delas se reflete na outra. Este é um conceito muito próximo ao de correlação, que estudaremos um pouco adiante (seção 2.1.2.5).

Tomando duas variáveis x e y de mesma dimensão, a sua covariância é definida de acordo com a equação (2.3) abaixo:

$$Cov_{x,y} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{2.3}$$

2.1.2.3 Desvio Padrão

O desvio padrão, definido como a raiz quadrada da variância, é a medida mais comum da dispersão estatística. Sua definição é realizada de forma a prover os estudiosos com uma medida da dispersão que seja um número não-negativo e que, também, use as mesmas unidades de medida dos dados observados. O desvio padrão, normalmente denotado por s, é expresso pela equação (2.4):

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.4}$$

Uma informação importante é que, para muitos conjuntos de dados, pode-se dizer que aproximadamente dois terços (ou 68%) das observações se encontram a uma distância da média inferior a um desvio padrão, aproximadamente 95% deles se encontram a uma distância de até duas vezes o desvio padrão e 99,7% a uma distância inferior a até três vezes o desvio padrão (BOWERMAN; O'CONNELL; KOEHLER, 2005). Esta informação é conhecida como a regra "68-95-99,7", que se aplica a conjuntos de dados com distribuição normal unimodal, simétrica, de afunilamento médio (ou mesocúrtica).

2.1.2.4 Estacionariedade

O conceito de estacionariedade é a base para a análise de séries temporais. Para fins de definição, podemos dizer que uma série temporal é estacionária se as suas propriedades estatísticas são independentes do período de tempo no qual são observadas (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 1998). No entanto, esta condição é bastante restritiva e difícil de ser verificada empiricamente e, por essa razão, uma definição menos restritiva de estacionariedade é normalmente utilizada: a estacionariedade fraca. Assim, dizemos que uma série X é fracamente estacionária se tanto sua média $\bar{x}$ quanto a covariância entre x_t e x_{t-l} sejam constantes no tempo, sendo l um inteiro qualquer. Segundo Tsay (2002), a estacionariedade fraca indica uma série que flutua com variação constante em torno de um nível fixo e é ela que nos possibilita realizar inferências relacionadas a observações futuras.

A covariância $\gamma_l = Cov(x_t, x_{t-l})$ é chamada autocovariância de *lag-l* de x_t (*lag* vêm do inglês, e significa "atraso") e possui duas importantes propriedades: $\gamma_0 = Var(x_t)$ e $\gamma_l = \gamma_{-l}$. A segunda propriedade é facilmente demonstrada, pois, $Cov(x_t, x_{t-(-l)}) = Cov(x_{t-(-l)}, x_t) = Cov(x_{t+l}, x_t) = Cov(x_t, x_{t_1-l})$, onde $t_1 = t + l$.

2.1.2.5 Autocorrelação

Em uma série fracamente estacionária, se a dependência linear entre o valor atual e os valores passados for de interesse, pode-se utilizar o conceito de autocorrelação. O coeficiente de correlação entre a variável x_t e x_{t-l} é chamado autocorrelação de *lag-l* de x_t, denotado normalmente como ρ_l e definido apenas em função de l pela equação (2.5) abaixo (TSAY, 2002). O importante neste conceito é o fato de que, se um elemento x_t da série apresentar uma correlação significativa com o elemento imediatamente anterior a ele, por exemplo, então este valor (chamado de *lag-1*) pode ser útil na previsão de x_t, servindo como uma das entradas e fornecendo informação relevante ao modelo previsor.

$$\rho_l = \frac{Cov(x_t, x_{t-l})}{\sqrt{Var(x_t)Var(x_{t-l})}} = \frac{Cov(x_t, x_{t-l})}{Var(x_t)} \tag{2.5}$$

Na equação, utiliza-se a propriedade $Var(x_t) = Var(x_{t-l})$, válida para estacionariedade fraca e, de sua definição, podemos dizer que $\rho_0 = 1$, $\rho_l = \rho_{-l}$, $-1 < \rho_l < 1$ e que a correlação entre x_t e x_{t-l} torna-se mais forte quando o coeficiente se afasta do valor zero. Assim, uma série fracamente estacionária não é serialmente correlacionada se e somente

se $\rho_l = 0$ para todo $l > 0$.

Finalmente, para um conjunto de amostras da série temporal estudada, podemos calcular seus coeficientes de autocorrelação através da equação (2.6):

$$\hat{\rho}_l = \frac{\Sigma_{t=l+1}^{T}(x_t - \bar{x})(x_{t-l} - \bar{x})}{\Sigma_{t=l+1}^{T}(x_t - \bar{x})^2}, \quad 0 \leq l < T - 1 \tag{2.6}$$

Os resultados obtidos através da aplicação da função de autocorrelação na série temporal podem ser expressos graficamente, através de um diagrama conhecido como correlograma (Figura 1). Segundo Bowerman, O'Connell e Koehler (2005), é possível também identificar a estacionariedade da série através da observação do correlograma seguindo algumas regras simples. Quando o coeficiente de correlação decresce rapidamente, ou simplesmente vai muito próximo a zero após ρ_0, a série pode ser considerada estacionária. Caso contrário, se os valores de autocorrelação diminuem lentamente no tempo, prolongando-se por muitos *lags*, a série pode ser considerada não-estacionária.

A Figura 1 nos mostra correlogramas, obtidos para representações distintas da informação contida na série de estudo desta dissertação: uma série financeira com valores das cotas de um fundo de investimento.

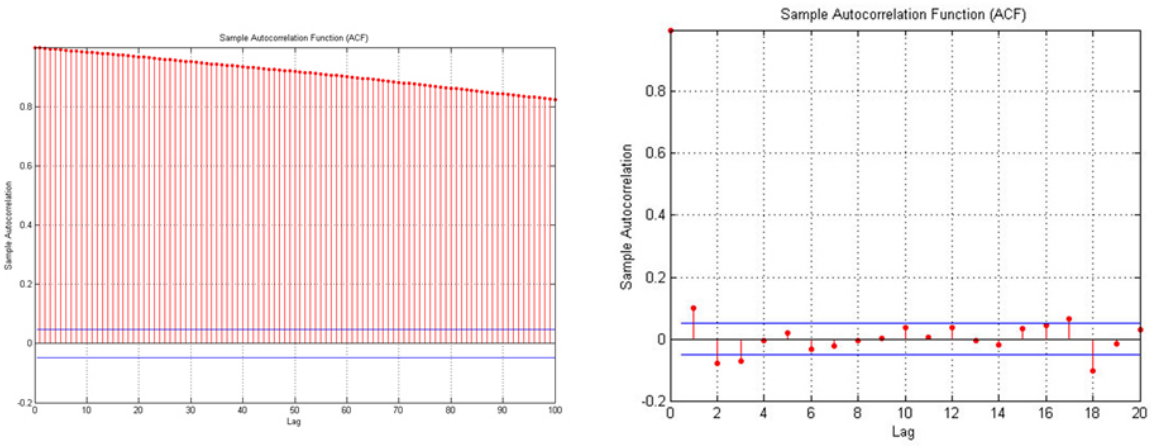

Figura 1: Autocorrelação: Série de preços / Série de retorno

No primeiro gráfico, a autocorrelação foi calculada com os dados da série original, sem normalização, contendo os preços das cotas do fundo. Através dos critérios introduzidos por Bowerman, O'Connell e Koehler (2005), verifica-se claramente a não-estacionariedade da série. Para facilitar o processo de previsão, a série passou por uma transformação, representada pelo cálculo da taxa de retorno diária do fundo de investimento, através da qual obteve-se uma série temporal estacionária de características estatísticas mais tratáveis

cujo correlograma está representado na mesma figura, à direita.

2.1.3 Série de Retorno

Na literatura financeira, é comum supor que a série de retorno de um ativo é fracamente estacionária (TSAY, 2002) e a grande maioria dos estudos financeiros envolve taxa de retorno ao invés de preço. Segundo Campbell, Lo e MacKinlay (1996), existem duas razões principais para sua utilização. A primeira delas é que o retorno de um ativo pode ser tratado como uma síntese completa e livre de escala da oportunidade de investimento. A segunda razão é que séries de retorno possuem características estatísticas mais atrativas que séries de preços, como a própria estacionariedade, que facilitam a análise e possibilitam a previsão. O retorno simples é calculado através da equação (2.7), onde P_t representa o preço e R_t a taxa de retorno do ativo no tempo t:

$$R_t = \frac{P_t}{P_{t-1}} - 1 \tag{2.7}$$

No caso desta dissertação, utilizou-se o retorno composto contínuo (ou *log return*), caracterizado pelo logaritmo natural do retorno simples bruto, que leva vantagem sobre o retorno simples pelo fato de ter algumas propriedades mais tratáveis. O retorno logarítmico é utilizado com muita frequência em pesquisas acadêmicas e sua vantagem principal é o fato de ser simétrico, diferente do retorno simples. Em termos práticos, pode-se dizer que em um investimento de R$100,00 que sofre uma queda com retorno igual a -50% seguida de uma alta com retorno de 50% resulta em um valor final de R$75,00 se utilizarmos o retorno simples e no mesmo valor de R$100,00 se o retorno composto contínuo for utilizado. Obviamente os resultados não são tão divergentes no caso da série estudada nesta dissertação, devido à pequena amplitude das mudanças percentuais diárias nos preços das cotas. O retorno contínuo é calculado através da equação (2.8):

$$R_t = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{2.8}$$

Tratando o retorno de um investimento como uma coleção de variáveis aleatórias no tempo nós obtemos uma série temporal, e a análise linear de séries temporais nos proporciona um *framework* natural para o estudo de sua dinâmica através de características que incluem estacionariedade, dependência dinâmica, função de autocorrelação, etc. A série original de preços (esquerda) e a série obtida com a aplicação da equação (2.8) (direita)

podem ser vistas na Figura 2.

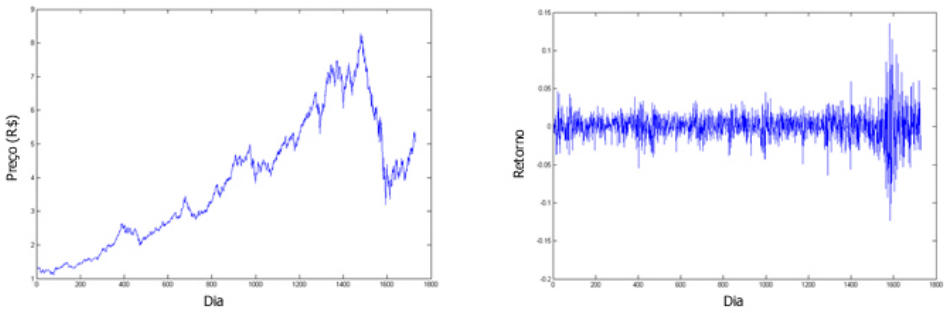

Figura 2: Séries de preço e retorno - Fundo de ações

A série de retorno, no que diz respeito às características discutidas no início do capítulo, termina por transformar a série não-estacionária inicial de preços, portadora de forte componente de tendência, em uma série estacionária de aspecto horizontal, com média próxima a zero (semelhante ao ruído branco) e algumas particularidades que nos levam a identificar um comportamento cíclico, caracterizado por períodos de maior ou menor estabilidade, de acordo com o aquecimento do mercado e o ciclo econômico natural.

2.1.4 Previsão em Séries Temporais

Em geral, existe um intervalo de tempo entre o conhecimento de um acontecimento iminente e sua ocorrência propriamente dita. É esse o tempo disponível para planejamento e definição de estratégias e é na antecipação do conhecimento e no aumento deste período de tempo disponível que entram as técnicas de previsão. A previsão em uma série temporal pode, portanto, ser definida como a estimação dos valores desta série em tempos futuros a partir de informação já disponível, objetivando diminuir a incerteza, otimizar ações e diminuir perdas.

Segundo Makridakis, Wheelwright e Hyndman (1998), existem cinco passos básicos a serem seguidos em qualquer tarefa de previsão:

1. **Definição do problema:** A primeira fase do projeto envolve a aquisição de um profundo conhecimento sobre as características intrínsecas do problema escolhido e a definição dos objetivos a serem perseguidos durante sua implementação. Neste passo, é necessário tomar conhecimento sobre os dados disponíveis para a formação da base de dados, informar-se sobre a relevância do problema e literatura referente à área, definir a maneira como o previsor será utilizado e os profissionais cujas necessidades serão supridas por ele.

2. **Obtenção das informações necessárias:** É sempre necessário realizar a coleta de dados históricos para auxiliar na construção (e treinamento) do modelo utilizado na previsão. Esta fase consiste, portanto, na aquisição dos dados numéricos a serem utilizados e, também, na obtenção de informações referentes à técnica escolhida para a elaboração do modelo (No caso deste trabalho, por exemplo, redes neurais e máquinas de aprendizagem).

3. **Análise preliminar dos dados:** Esta fase é responsável pela identificação das características da série estudada e seu pré-processamento, com o objetivo de maximizar o desempenho do modelo previsor. Aqui são computados dados estatísticos descritivos (média, variância) e identificados os componentes (sazonal, tendência) e características importantes (estacionariedade) da série estudada. Por fim, pode-se realizar transformações nos dados de forma que se adequem melhor ao conjunto de ferramentas disponível e ao modelo previsor.

4. **Definição e ajuste do modelo:** O penúltimo passo do processo consiste na escolha do modelo mais adequado ao problema apresentado e investigado nos passos anteriores e no ajuste de seus parâmetros para a obtenção de um resultado ótimo. Aqui são explorados os detalhes técnicos da implementação, resolvidos os problemas práticos e definido o modelo utilizado e avaliado na parte final.

5. **Utilização e avaliação do modelo:** Finalmente o modelo concebido na fase anterior passa pelo estágio de avaliação de desempenho, sendo utilizado para realizar a previsão em um conjunto de dados reais retirado especialmente para o teste. O avaliador define a metodologia utilizada, com o objetivo de obter uma análise imparcial dos resultados e a avaliação final da eficiência do modelo. Algumas métricas para avaliação de resultados serão apresentadas na seção a seguir (2.1.5).

No decorrer da pesquisa os passos para uma previsão confiável foram seguidos e estão distribuídos pelos capítulos desta dissertação. Os capítulos 1 e 2 apresentam os caminhos percorridos para a realização dos três primeiros passos. A definição das características do modelo utilizado se encontram no capítulo 3. Por fim, a obtenção dos resultados e posterior avaliação se encontram distribuídos pelos capítulos 4 e 5.

2.1.5 Medidas de Eficiência

Para que se tenha uma idéia clara e realista da eficiência de um modelo previsor é necessário estabelecer uma medida confiável da magnitude do erro obtido na previsão.

Através da adoção de uma medida padrão para descrição do erro, é possível verificar se o modelo estudado é adequado para o problema que se deseja tratar e, ainda, estabelecer uma escala para a comparação entre diferentes modelos ou arquiteturas no tratamento do mesmo problema.

Tomando a definição intuitiva de erro de previsão, uma das medidas de erro mais simples é a do "erro médio", chamado de ME (do inglês, *mean error*) e definido pela equação (2.9) abaixo. Apesar de sua simplicidade e aparente precisão, o erro ME não é uma medida confiável e tende a ser pequeno, na maioria dos casos, devido à possibilidade de ocorrência de erros positivos e negativos, que acabam muitas vezes por se anularem durante o somatório.

$$ME = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{2.9}$$

A solução para o problema do erro médio se encontra na utilização de outras duas medidas: o "erro absoluto médio" MAE (*mean absolute error*) e o "erro quadrático médio" MSE (*mean squared error*). O erro MAE se livra do problema apresentado pelo ME tomando apenas o valor absoluto dos erros obtidos, enquanto o MSE, com abordagem similar, eleva os erros ao quadrado com o mesmo objetivo: a eliminação do sinal. As fórmulas para o cálculo do MAE e do MSE são representadas respectivamente pelas equações (2.10) e (2.11) abaixo. As duas medidas possuem vantagens e desvantagens. Enquanto o MSE se beneficia por ser matematicamente mais simples de ser manipulado, o MAE possui a vantagem de uma interpretação mais direta por parte do analista.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.10}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.11}$$

Um fato a ser destacado como ponto falho nas três medidas apresentadas até então é que a magnitude alcançada por elas depende diretamente da escala dos dados, dificultando a análise dos resultados fora de um contexto mais elaborado. Apenas como fator ilustrativo, podemos dizer, por exemplo, que um erro de 1 litro na previsão mensal de produção de uma bebida em uma fábrica que produz 100 litros mensais é muito diferente do mesmo erro em uma previsão para uma fábrica que produza 100 mil litros por mês. Por essa razão, a utilização de medidas percentuais - e portanto, sem escala - é mais indicada

na avaliação de modelos previsores.

Dois erros percentuais bem difundidos são o "erro percentual médio" MPE (*mean percentage error*) e o "erro percentual absoluto médio" MAPE (*mean absolute percentage error*), cujas fórmulas estão expressas abaixo nas equações (2.12) e (2.13). Uma das desvantagens tanto do MPE quanto do MAPE é que ambos apresentam problemas quando o valor y_i for igual a 0, devido à sua utilização no denominador das equações. O MPE apresenta ainda os mesmos problemas relativos ao cancelamento de erros de sinais opostos do erro médio ME.

$$MPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right) \tag{2.12}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{2.13}$$

Apesar da desvantagem citada, o MAPE se apresenta como uma medida mais clara e intuitiva, facilitando a interpretação dos resultados e comparação entre métodos. Por exemplo, dizer que um experimento resultou em um MAPE de 5% é muito mais significativo do que informar que o MSE, ou erro quadrático médio, foi de 21. O erro percentual absoluto médio apresenta-se, portanto, como uma medida livre de escala e ideal para a avaliação da eficiência de um modelo de previsão.

É necessário salientar que, ainda que uma escolha tenha sido tomada, nenhuma das medidas de erro apresentadas fornece, por si só, fundamentos suficientes para a avaliação de um método de previsão. Enquanto aplicações menos exigentes (ou extremamente complexas) podem apontar um erro MAPE de 10% como um bom resultado de previsão, em outras, um erro de 2,5% utilizando a mesma métrica pode significar um verdadeiro desastre.

A solução para este impasse se encontra na utilização de valores de referência para comparação. Resultados de testes com um modelo recém-concebido devem ser sempre apresentados ao lado de resultados obtidos por modelos mais simples ou de igual complexidade, mas já estabelecidos no mercado ou no ambiente acadêmico, e utilizados com frequência para a solução do mesmo problema. A comparação entre os novos resultados e os demais, alcançados por outros modelos, é que poderá definir se o modelo é eficiente e qual deles, dentre os estudados, apresenta melhor desempenho.

2.2 Redes Neurais

Desde que a comunidade científica deu seus primeiros passos em direção ao desenvolvimento dos computadores, já existia a idéia da construção de uma máquina inteligente, capaz de se comportar como um ser humano e resolver problemas de maneira semelhante ao nosso cérebro. O conceito de uma máquina provida de inteligência acabou por se tornar parte do imaginário popular através, por exemplo, das leis da robótica de Isaac Asimov (ASIMOV, 1963) e do robô humanóide C-3PO concebido por George Lucas para sua saga de ficção "Guerra nas Estrelas".

No entanto, é necessário admitir que, apesar de se tratarem de unidades de processamento, existem muitas diferenças entre um computador e o cérebro humano. Seres humanos possuem aptidões, como o reconhecimento de objetos pela visão e de sons familiares através da audição, que não são de fácil solução para o computador. Especialmente, pode-se dizer que somos capazes de solucionar problemas de múltiplas variáveis através de conhecimento prévio, de experiência adquirida com a vivência. Computadores, por sua vez, possuem grande vantagem na realização de cálculos complexos, localização de registros e armazenamento de dados, que não são compartilhados pelo ser humano.

Um dos problemas para o desenvolvimento de uma técnica que simule o comportamento do cérebro humano é que muito pouco ainda se sabe sobre os aspectos de seu funcionamento e estrutura. Hawkins e Blakeslee (2004) destacam que existem muitos livros explicando passo a passo os mais diversos assuntos - desde a teoria da relatividade, buracos negros até teorias sobre a extinção dos dinossauros - mas não sobre o funcionamento detalhado do cérebro. Sabe-se, entretanto, que o cérebro humano é formado por inúmeros componentes, conhecidos como neurônios, que formam uma estrutura complexa e paralelizada. Conhece-se também, que sua arquitetura é dividida em áreas dedicadas, que possibilitam a realização de diversas tarefas simultaneamente (HAYKIN, 2001). Além disso, podemos dizer que o cérebro utiliza um modelo baseado em memória para a realização de contínuas previsões de eventos futuros e, segundo Hawkins e Blakeslee (2004), é nessa habilidade que consiste sua inteligência.

Assim, podemos dizer que uma das grandes motivações do trabalho e pesquisa no campo das redes neurais é biológica: o fato de que o cérebro humano processa informações de forma completamente diferente de um computador digital. O cérebro humano é uma central de processamento altamente complexa, não-linear e paralela. As redes neurais artificiais se beneficiam das características deste tipo de processamento, sendo capazes de

concluir tarefas complexas devido à sua arquitetura maciçamente paralela e distribuída e de sua capacidade de aprendizado e generalização.

As redes neurais são, portanto, sistemas paralelos distribuídos compostos por unidades de processamento simples - chamadas nós computacionais ou neurônios - dispostas em uma ou mais camadas e interligadas por um grande número de conexões, comumente chamadas de sinapses. O processo de solução de um problema passa inicialmente por uma fase de aprendizagem, na qual um conjunto significativo de padrões é apresentado à rede, que extrai as características necessárias para representar a informação fornecida. Essa informação é geralmente armazenada em valores atribuídos às conexões sinápticas, os quais chamamos de pesos.

A capacidade de generalização mencionada anteriormente se refere ao fato de que a rede neural deve produzir saídas adequadas para entradas que não estavam presentes durante o treinamento (HAYKIN, 2001). Através dessa capacidade é possível solucionar complexos problemas de classificação, reconhecimento de padrões e regressão.

Na prática, contudo, as redes neurais não fornecem soluções para problemas de alto nível de complexidade trabalhando individualmente. Em vez disso, elas precisam ser integradas em uma abordagem consistente de engenharia de sistemas. Especificamente, um problema complexo de interesse é decomposto em um número de tarefas relativamente simples, e atribui-se a redes neurais um subconjunto de tarefas que coincidem com suas capacidades inerentes, de maneira semelhante ao realizado pelas redes neurais biológicas. Finalmente, apesar de todos os avanços, é importante reconhecer que nós temos um longo caminho a percorrer antes de contruirmos (se porventura conseguirmos) uma arquitetura computacional que mimetize o cérebro humano (HAYKIN, 2001).

A utilização de redes neurais proporciona um grande número de benefícios através de suas propriedades e capacidades inerentes. Algumas das mais importantes delas são:

1. **Não-Linearidade:** Um rede construída a partir de neurônios não-lineares é, também, não-linear e sua não-linearidade é de um tipo especial, por ser distribuída pela rede. A não-linearidade é uma propriedade muito importante, particularmente se o sinal de entrada possuir natureza essencialmente não-linear.

2. **Mapeamento de Entrada-Saída:** A rede neural é capaz de aprender através de exemplos e armazenar o conhecimento nas conexões sinápticas, de maneira a mapear padrões de entrada para saídas (ou resultados) apropriadas.

3. **Adaptabilidade:** As redes neurais possuem a capacidade de adequar seus pesos

sinápticos a modificações no meio ambiente, ou seja, uma rede treinada para operar em um ambiente específico pode ser facilmente retreinada para lidar com pequenas modificações nas condições operativas do ambiente. Além disso, quando operando em um ambiente não-estacionário, a rede pode ser projetada especialmente para que sua arquitetura seja capaz de mudar os pesos sinápticos em tempo real.

4. **Informação Contextual:** O conhecimento é representado pelo próprio estado de ativação da rede, além de sua estrutura. Informação de contexto é tratada, portanto, com naturalidade pelas redes neurais e pode ser mais explorada com a utilização de arquiteturas especialmente projetadas para esse fim.

5. **Uniformidade de Análise e Projeto:** No domínio de redes neurais, a mesma notação é utilizada em praticamente todas as linhas de pesquisa. Como exemplo, podemos tomar o conceito de neurônio, que é comum a todos os tipos de rede. Essa uniformidade torna possível o compartilhamento de teorias e algoritmos de aprendizagem em diferentes aplicações de redes neurais e facilita, também, a construção de redes modulares.

Existem várias arquiteturas e modelos de redes neurais artificiais, cada um com suas próprias características e fins. De maneira a tornar claro o desenvolvimento do trabalho, as próximas seções exploram os conceitos envolvidos nos três tipos de redes utilizadas: os perceptrons de múltiplas camadas (MLP, *multi-layer parceptron*), os mapas auto-organizáveis (SOM, *self-organizing maps*) e as máquinas de vetor de suporte (SVM, *support vector machines*).

2.2.1 MLP - Perceptron de Múltiplas Camadas

2.2.1.1 Perceptron de Camada Única

Nos primeiros anos das redes neurais, especialmente no período entre 1943 e 1958, vários pesquisadores se destacaram por suas contribuições pioneiras e entre eles se sobressai Rosenblatt (1958), pela proposição do perceptron como primeiro modelo neural para aprendizagem supervisionada, ou seja, com auxílio de um supervisor.

O perceptron é a forma mais simples de uma rede neural e pode ser utilizado somente para a classificação de padrões ditos linearmente separáveis, ou seja, padrões que se encontrem em lados opostos de um hiperplano. Basicamente, ele consiste em um único

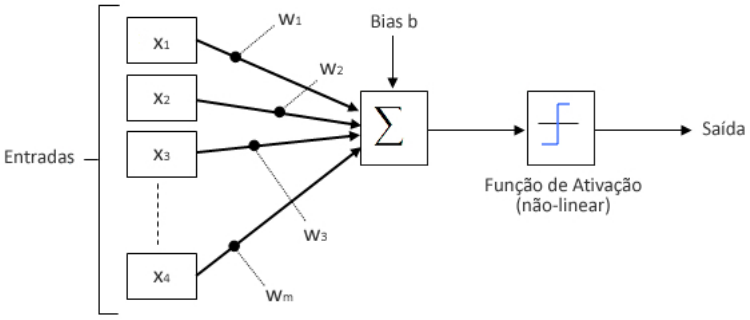

Figura 3: Modelo de neurônio de McCulloch-Pitts

neurônio com pesos ajustáveis e bias, sendo este neurônio não-linear, de acordo com o modelo McCulloch-Pitts (MCCULLOCH; PITTS, 1943) ilustrado na Figura 3.

Este modelo de neurônio consiste em um combinador linear seguido por um limitador abrupto, realizando o papel de função de ativação não-linear. Da figura, observamos que o nó aditivo calcula uma combinação linear das entradas aplicadas às sinapses do neurônio e incorpora um bias aplicado externamente. A soma resultante, também chamada de **campo local induzido**, é então aplicada ao limitador abrupto resultando em uma saída igual a +1 se a entrada do limitador for positiva e -1 caso contrário. O campo local induzido no neurônio é calculado através equação (2.14), e o limitador, representado pelo símbolo $sinal(.)$, obtido pela equação (2.15).

$$\nu = \sum_{i=1}^{m} w_i x_i + b \tag{2.14}$$

$$sinal(\nu) = \begin{cases} +1 \ se \ \nu \geq 0 \\ -1 \ se \ \nu < 0 \end{cases} \tag{2.15}$$

O objetivo do perceptron é classificar corretamente o conjunto de estímulos aplicados externamente x_1, x_2, ... x_m em uma de duas classes (HAYKIN, 2001), representadas pelas saídas +1 e -1 e separadas por um hiperplano definido pela equação (2.16) abaixo.

$$\sum_{i=1}^{m} w_i x_i + b = 0 \tag{2.16}$$

Para o caso de um problema bidimensional, a fronteira de decisão estabelecida pelo hiperplano toma forma de uma linha reta. Os pontos que se encontram de um lado da reta são classificados como pertencentes a uma classe e os que estão do lado contrário como pertencentes à segunda classe. Através da Figura 4, observamos este comportamento e

identificamos a utilidade do bias: deslocar a fronteira de decisão em relação à origem.

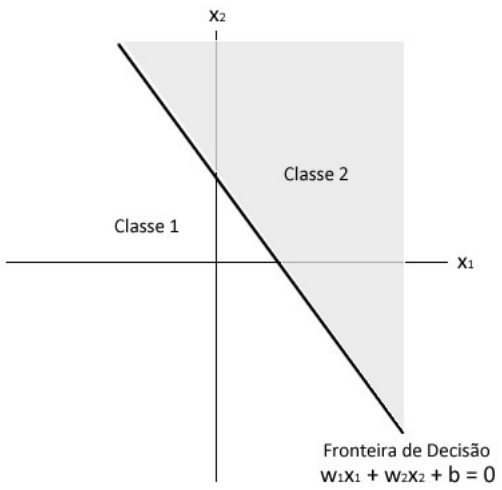

Figura 4: Hiperplano como fronteira de decisão em problema bidimensional

Os pesos sinápticos do perceptron podem ser adaptados durante o processo de aprendizagem através da utilização de uma regra de correção de erro conhecida como algoritmo de convergência do perceptron. Para explicar o seu funcionamento e tendo em mente que n representa o instante no tempo, precisamos deixar claras algumas definições:

- $\mathbf{x}(n)$ é o vetor de entrada $[+1, x_1(n), x_2(n), ..., x_m(n)]^T$;
- $\mathbf{w}(n)$ é o vetor de pesos $[b(n), w_1(n), w_2(n), ..., w_m(n)]^T$;
- $b(n)$ é o nível de bias;
- $y(n)$ é a resposta real, obtida pelo modelo;
- $d(n)$ é a resposta desejada;
- η é um parâmetro conhecido como taxa de aprendizagem (constante positiva, menor que a unidade).

Os passos para o algoritmo de convergência do perceptron, segundo Haykin (2001), são:

1. **Inicialização:** Faz-se $\mathbf{w}(0) = \mathbf{0}$. A partir daí, deve-se executar os passos 2 a 5 repetidamente nos passos de tempo $n = 1, 2, ...$

2. **Ativação:** No passo de tempo n, ativa-se o perceptron aplicando o vetor de entrada $\mathbf{x}(n)$ e a resposta desejada $d(n)$.

3. **Cálculo da Resposta Real:** Calcula-se a resposta real do perceptron, através da equação (2.17):

$$y(n) = sinal[\mathbf{w}^T(n)\mathbf{x}(n)] \tag{2.17}$$

4. **Adaptação do Vetor Peso:** Finalmente atualiza-se o vetor peso do perceptron, de acordo com a fórmula:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n) \tag{2.18}$$

Onde:

$$d(n) = \begin{cases} +1 \ se \ \mathbf{x}(n) \in Classe \ 1 \\ -1 \ se \ \mathbf{x}(n) \in Classe \ 2 \end{cases} \tag{2.19}$$

5. **Continuação:** Incrementa-se o passo de tempo n e retorna-se para o passo 2.

Os passos de 2 até 4 são repetidos até que a saída apresente resultados dentro de uma tolerância esperada, ou seja, até que os elementos sejam classificados corretamente.

Da equação (2.18), podemos dizer que a diferença $d(n) - y(n)$ assume o papel de sinal de erro. O parâmetro da taxa de aprendizagem é uma constante positiva restrita ao intervalo $0 < \eta \leq 1$ e, ao atribuir valores nesse intervalo, devemos considerar dois requisitos conflitantes: valores pequenos de η fornecem estimativas estáveis para o peso, enquanto valores grandes geram adaptações mais rápidas. Por este motivo, é importante que o parâmetro η seja selecionado cuidadosamente, para assegurar que seja alcançada a estabilidade ou convergência do processo de aprendizagem iterativo.

A limitação deste modelo consiste no fato de ser aplicável somente em problemas com padrões linearmente separáveis, o que praticamente encerrou os estudos na área de redes neurais, gerando uma época de poucas novidades entre as décadas de 60 e 80. O limitador abrupto constitui o elemento não-linear do neurônio de McCulloch-Pitts e, apesar da intuição nos indicar o contrário, a utilização de uma não-linearidade mais suave (sigmóide, por exemplo) não levaria à solução de problemas com padrões que não sejam linearmente separáveis: as características de regime permanente de tomada de decisão, de estado estável do perceptron, são basicamente sempre as mesmas, não importando se é utilizado

um limitador abrupto ou um limitador suave como fonte de não-linearidade do modelo neural. Somente com a proposta do modelo Perceptron de Múltiplas Camadas (MLP, do inglês *multi-layer perceptron*) e de um novo algoritmo de treinamento (a retropropagação de erro) é que surgiram avanços significativos para a continuação das pesquisas.

2.2.1.2 Perceptron de Múltiplas Camadas

Com a finalidade de se contornar as limitações e servir como ampliação do modelo perceptron de camada única surgiu o perceptron de múltiplas camadas. Nesse novo modelo, também baseado no processo de aprendizagem supervisionada, existem três conjuntos de unidades computacionais: os nós de entrada (também chamados nós sensoriais ou nós de fonte) que constituem uma camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída. Trata-se de uma rede alimentada adiante (*feedforward*), pois o sinal se propaga para frente, camada a camada. Uma representação básica de sua estrutura está ilustrada na Figura 5.

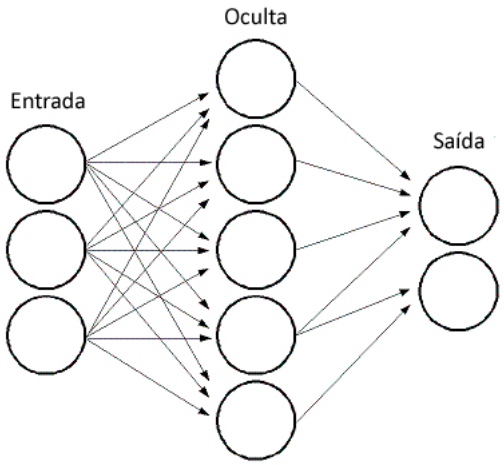

Figura 5: Representação básica do perceptron de múltiplas camadas

Segundo Haykin (2001), existem três características particulares do perceptron de múltiplas camadas que o distinguem dos demais modelos. É através dessas características, em conjunto com a habilidade de aprender da experiência através do treinamento, que o modelo deriva seu poder computacional. São elas:

1. O modelo de cada neurônio da rede inclui uma função de ativação não-linear, com não-linearidade suave, diferenciável em qualquer ponto, diferente da limitação abrupta utilizada no perceptron de Rosenblatt. Uma das funções mais utilizadas a se enquadrarem nessa condição é a sigmóide, caracterizada pela função logística:

$$y_j = \frac{1}{1 + e^{-\nu_j}} \tag{2.20}$$

Onde ν_j é o campo local induzido do neurônio j, calculado através da soma de todas as entradas sinápticas, e y_j é a saída do mesmo neurônio.

2. A rede possui uma ou mais camadas de neurônios ocultos, responsáveis pela extração gradual das características mais significativas presentes nos padrões de entrada durante o processo de treinamento.

3. O modelo exibe ainda um alto grau de conectividade, determinado pelas sinapses da rede. Uma modificação na conectividade da rede requer uma mudança na população das conexões sinápticas e de seus pesos.

O segundo pilar das redes MLP consiste em seu algoritmo de treinamento supervisionado baseado na regra de aprendizagem por correção de erro, chamado de algoritmo de **retropropagação de erro** (popularmente conhecido como *error back-propagation*, do inglês). O funcionamento do algoritmo consiste, basicamente, em dois passos: a propagação (para frente) e a retropropagação (para trás). No primeiro passo, um padrão é aplicado aos nós de entrada da rede e seu efeito se propaga através do modelo, camada por camada, até gerar um sinal de saída como resposta da rede. Durante a primeira etapa os pesos sinápticos são fixos. Na segunda parte do algoritmo - a retropropagação - é que os pesos são ajustados, de acordo com uma regra de correção de erro. Nela, a resposta obtida pelo modelo é subtraída da resposta desejada produzindo um sinal de erro. Este sinal de erro é então propagado para trás através da rede, contra a direção das conexões sinápticas. Os pesos são então ajustados de maneira a fazer com que a resposta real da rede se aproxime cada vez mais da resposta desejada, em um sentido estatístico. O funcionamento do algoritmo será detalhado adiante, na seção 2.2.1.3. As duas etapas do algoritmo de retropropagação estão ilustradas de forma simplificada na Figura 6.

2.2.1.3 Algoritmo de Retropropagação de Erro

O algoritmo de retropropagação de erro para o treinamento do perceptron de múltiplas camadas possui alguns passos básicos que devem ser seguidos para o seu funcionamento correto. Segundo Haykin (2001), os passos (de maneira resumida) são:

1. **Inicialização:** Iniciam-se os pesos sinápticos e limiares (valor limite para ativação

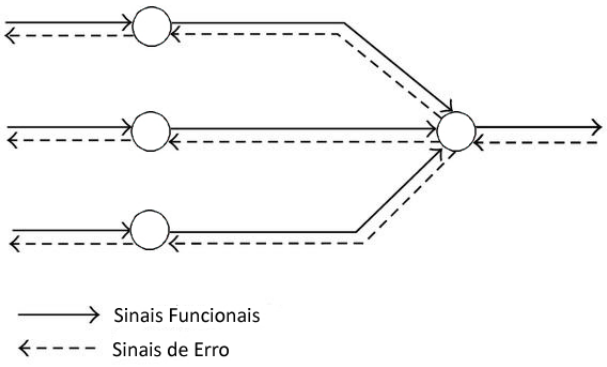

Figura 6: Passos do algoritmo de retropropagação de erro

do neurônio) com valores pequenos e aleatórios, retirados de uma distribuição uniforme.

2. **Apresentação dos exemplos de treinamento:** Apresenta-se uma época de exemplos de treinamento à rede. Para cada exemplo do conjunto devem ser realizadas as computações descritas nos passos 3 e 4.

3. **Propagação:** Toma-se $(\mathbf{x}(n), \mathbf{d}(n))$ como um exemplo de treinamento da época, onde $\mathbf{x}(n)$ é o vetor de dados aplicado à camada de entrada e $\mathbf{d}(n)$ o vetor resposta, apresentado à camada de saída da rede. Calcula-se, então, os campos locais induzidos e os sinais funcionais da rede prosseguindo para frente, camada a camada. O campo local induzido $\nu_j^{(l)}(n)$ para o neurônio j na camada l é dado pela equação (2.21), abaixo:

$$\nu_j^{(l)}(n) = \sum_{i=0}^{m} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \tag{2.21}$$

Onde $y_i^{(l-1)}(n)$ é o sinal da função de saída do neurônio i na camada anterior $(l-1)$, na iteração n, $w_{ji}^{(l)}(n)$ é o peso sináptico do neurônio j da camada l, que é alimentado pelo neurônio i da camada $l-1$, e m é o número total de entradas aplicadas ao neurônio j. Assumindo-se a utilização da função sigmóide (φ), o sinal de saída do neurônio j na camada l é dado por 2.22:

$$y_j^{(l)} = \varphi_j(\nu_j^{(l)}(n)) \tag{2.22}$$

Se o neurônio j for da primeira camada oculta $(l = 1)$, as entradas aplicadas ao neurônio serão:

$$y_j^{(0)} = x_j(n) \tag{2.23}$$

Onde $x_j(n)$ é o j-ésimo elemento do vetor de entrada $\mathbf{x}(n)$. Já se o neurônio for da camada de saída (chamemos de L, onde L é a profundidade da rede), determina-se os valores de saída $o_j(n)$:

$$y_j^{(L)} = o_j(n) \tag{2.24}$$

Por último, calcula-se o sinal de erro, através da função 2.25:

$$e_j(n) = d_j(n) - o_j(n) \tag{2.25}$$

Onde $d_j(n)$ é o j-ésimo elemento do vetor resposta desejada $\mathbf{d}(n)$.

4. **Retropropagação:** Calcula-se os gradientes locais δ da rede, definidos por:

$$\delta_j^{(l)}(n) = e_j^{(L)}(n)\varphi_j'(\nu_j^{(L)}(n)) \tag{2.26}$$

caso o neurônio j for da camada de saída L e:

$$\delta_j^{(l)}(n) = \varphi_j'(\nu_j^{(l)}(n)) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \tag{2.27}$$

caso o neurônio j seja da camada oculta l. Nas equações, o apóstrofe (como em $\varphi_j'(.)$) representa a diferenciação em relação ao argumento. Por último, ajusta-se os pesos sinápticos de acordo com a regra delta generalizada:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[\Delta w_{ji}^{(l)}(n-1)] + \eta\delta_j^{(l)}(n) y_i^{l-1}(n) \tag{2.28}$$

Onde η é a taxa de aprendizagem e α é a constante de momento, escolhidas pelo usuário. Estes parâmetros são normalmente ajustados (reduzidos) quando o número de iterações aumenta.

5. **Iteração:** Realiza-se iterações das computações realizadas nos passos 3 e 4, apresentando novas épocas de exemplos de treinamento para a rede, até que seja satisfeito o critério de parada. A ordem de apresentação dos exemplos de treinamento deve ser aleatória, de época para época.

Não é possível demonstrar a convergência do algoritmo de retropropagação, mas existem critérios de parada que podem ser utilizados para encerrar a fase de ajuste dos pesos. Por exemplo, pode-se estabelecer que o ajuste cessa quando a taxa absoluta de variação do erro médio quadrado por época for suficientemente pequena. Existe ainda um critério útil e teoricamente fundamentado chamado "validação cruzada". Nele, após cada iteração do processo de aprendizagem, a rede é testada pelo seu desempenho de generalização em um conjunto separado de dados: o conjunto de validação. Caso o desempenho seja adequado ou atinja seu máximo, o processo é encerrado.

2.2.1.4 Considerações Gerais

A aprendizagem por retropropagação emergiu como o algoritmo padrão de treinamento de perceptrons de múltiplas camadas, especialmente a partir da publicação do livro de Rumelhart e McClelland (1986), *Parallel Distributed Processing*, e serve como modelo de comparação para outros algoritmos de aprendizagem. No entanto, a presença de uma forma distribuída de não-linearidade a a alta conectividade da rede tornam muito difícil a sua análise teórica (HAYKIN, 2001).

Os neurônios ocultos desempenham um papel fundamental na operação do modelo, atuando como detectores de características. Conforme o processo de aprendizagem avança, são eles os responsáveis pela descoberta de características salientes que definem o conjunto de treinamento. Essa tarefa é realizada através de uma transformação não-linear nos dados de entrada para um novo espaço chamado de espaço oculto (ou espaço de características), onde a tarefa a ser executada pela rede pode ser mais facilmente cumprida. Portanto, a utilização desses neurônios ocultos torna o processo de aprendizagem mais eficiente, mas também mais difícil de ser visualizado.

Um problema recorrente em diversos modelos de redes neurais artificiais é o excesso de ajuste ou excesso de treinamento (*over-fitting*, no inglês). Dizemos que uma rede é eficiente quando ela produz um mapeamento de entrada-saída correto, mesmo quando a entrada for um pouco diferente dos padrões utilizados para treinamento. Esse fenômeno é conhecido como generalização. Entretanto, quando o processo de aprendizagem se prolonga por um número excessivo de épocas de treinamento, a rede pode acabar memorizando os dados de treinamento. Quando a rede é excessivamente treinada ela se torna incapaz de generalizar devido, por exemplo, ao aprendizado de características que se encontram presentes nos dados (como ruído), mas não na função subjacente que deve ser modelada. A generalização, portanto, não deve ser tratada como uma propriedade mítica das redes

neurais, mas simplesmente como o efeito de uma boa interpolação não-linear sobre os dados de entrada.

Além disso, um dos problemas mais graves do perceptron de múltiplas camadas e que também deve ser levantado é a presença de mínimos locais na superfície de erro. Como a aprendizagem por retropropagação de erro é basicamente uma técnica de "escalada de colina", ela corre o risco de ficar presa em um mínimo local, onde toda variação dos pesos sinápticos - por menor que seja - causa aumento na função de custo. Esta característica pode levar a um alto valor de erro, pois, em algum outro lugar do espaço de pesos pode existir outro conjunto de pesos sinápticos para o qual a função de custo é menor que a obtida no mínimo local onde a rede se encontra presa. É evidentemente indesejável que o processo de treinamento termine em um mínimo local - especialmente se estiver muito longe do mínimo global - e este fator deve ser sempre levado em consideração pelo usuário da rede.

2.2.2 SOM - Mapas Auto-Organizáveis

2.2.2.1 Estrutura do Mapa

A inspiração biológica para o desenvolvimento de novos modelos neurais artificiais é uma constante durante a história desse campo de pesquisa. Para o caso dos mapas auto-organizáveis (KOHONEN, 1990), uma importante característica do cérebro humano se mostrou fundamental: o fato de que o cérebro se encontra organizado de modo que entradas sensoriais diferentes (motora, visual, auditiva, etc.) sejam mapeadas para áreas diferentes do córtex cerebral, de uma maneira topologicamente ordenada. Assim, cada parte da informação recebida pelo sistema complexo é mantida no seu próprio contexto e neurônios especializados em cada uma delas podem interagir entre si através de conexões sinápticas curtas. Segundo Haykin (2001), é por este motivo que a localização espacial de um neurônio de saída em um mapa topográfico corresponde a um domínio ou característica particular do dado retirado do espaço de entrada.

Os mapas auto-organizáveis (SOM, do inglês *self-organizing maps*) são redes neurais alimentadas adiante que utilizam um algoritmo de treinamento não-supervisionado e, através de um processo chamado auto-organização, configuram suas unidades de saída em uma representação topológica dos dados originais (DEBOECK, 1998). Nas redes SOM, as localizações espaciais dos neurônios de saída indicam o grau de semelhança entre diferentes padrões de entrada.

A rede SOM é formada por uma camada de nós sensoriais de entrada conectados aos nós da camada de saída, dispostos geralmente em uma grade bidimensional. Esta grade representa uma estrututra alimentada adiante com uma única camada computacional consistindo de neurônios arranjados em linhas e colunas, cada um deles conectado a todos os nós de fonte da camada de entrada. O modelo está ilustrado na Figura 7.

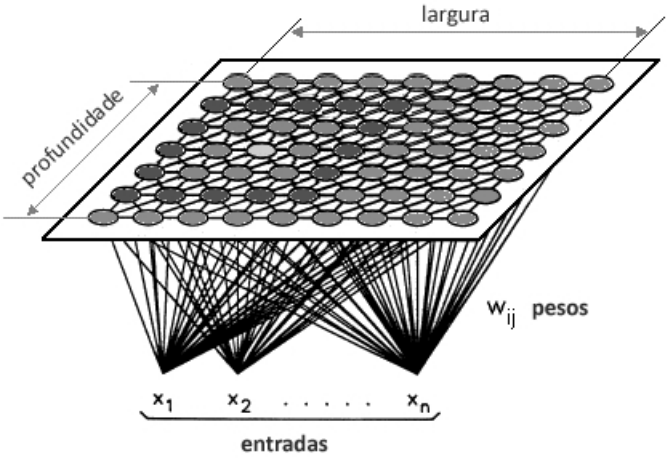

Figura 7: Mapa auto-organizável de Kohonen

Na camada de saída, os neurônios se tornam seletivamente sintonizados a vários padrões de entrada (estímulos) ou classes de padrões de entrada no decorrer do processo de aprendizagem. Estas grades são baseadas na aprendizagem competitiva, onde os neurônios de saída competem entre si para serem ativados de modo que apenas um deles é ativado num determinado instante de tempo (*winner-takes-it-all*). A cada entrada os pesos das conexões são adaptados com a finalidade de criar na grade um sistema significativo de coordenadas, que representem fielmente diferentes características da entrada.

2.2.2.2 O Algoritmo SOM

O algoritmo responsável pelo treinamento de um mapa auto-organizável consiste em algumas etapas bem definidas:

1. **Inicialização:** Atribui-se valores pequenos e aleatórios aos pesos sinápticos da rede, de maneira que nenhuma organização inicial seja imposta ao mapa de características.

2. **Competição:** Nessa fase os neurônios competem entre si pela ativação. Para cada padrão de entrada, todos os neurônios calculam os valores de uma função discriminante. O neurônio que obtiver o maior valor vence e é declarado vencedor (*winner-takes-it-all*).

Consideremos $\mathbf{x}$ um padrão de entrada selecionado aleatoriamente do espaço de entrada, e $\mathbf{w}_j$ como o vetor peso sináptico de cada neurônio j ($j = 1, 2, ..., l$) da grade, com a mesma dimensão do espaço de entrada. Para encontrar o melhor casamento entre o vetor de entrada e pesos sinápticos, deve-se calcular os produtos internos $\mathbf{w}_j^T\mathbf{x}$ para cada um dos neurônios da grade e selecionar o de maior resultado. No entanto, a maximização do produto $\mathbf{w}_j^T\mathbf{x}$ é matematicamente equivalente à minimização da distância euclidiana entre os vetores $\mathbf{x}$ e $\mathbf{w}_j$, quando $\mathbf{w}_j$ for normalizado. Finalmente, utilizando $i(\mathbf{x})$ para representar o neurônio vencedor, podemos determiná-lo através da equação (2.29), abaixo:

$$i(\mathbf{x}) = \arg\min_j ||\mathbf{x} - \mathbf{w}_j||, \;\; j = 1, 2, ..., l \tag{2.29}$$

Assim, um espaço contínuo de entrada de padrões de ativação é mapeado para um espaço discreto de saída de neurônios por um processo de competição entre os neurônios da grade (HAYKIN, 2001).

3. **Cooperação:** O neurônio vencedor da etapa de competição define a localização de uma vizinhança topológica de neurônios que também devem se excitar com o padrão de entrada, fornecendo a base para a cooperação entre neurônios vizinhos.

   Consideremos que $d_{j,i}$ represente a distância lateral entre o neurônio vencedor i e o neurônio excitado j, e que $h_{j,i}$ represente a vizinhança topológica centrada no neurônio vencedor i e que contenha um conjunto de neurônios excitados j. Podemos dizer que a função $h_{j,i}$ deve alcançar seu valor máximo no neurônio vencedor i ($d_{i,j} = 0$) e que sua amplitude deve decrescer monotonicamente com o aumento dessa distância, decaindo para 0 quando a distância tender para infinito. Um exemplo de escolha típica de $h_{j,i}$, portanto, é a função gaussiana, representada na equação (2.30) abaixo e ilustrada na Figura 8.

$$h_{j,i(x)} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right) \tag{2.30}$$

   Onde o parâmetro σ é o raio da gaussiana, ou a "largura efetiva" da vizinhança topológica, que mede o grau com o qual os neurônios excitados na vizinhança do neurônio vencedor participam do processo de aprendizagem. Sua utilização é biologicamente mais apropriada do que uma vizinhança retangular, por exemplo, mas a função a ser utilizada deve ser escolhida através de testes, de acordo com a sua eficiência no contexto de interesse.

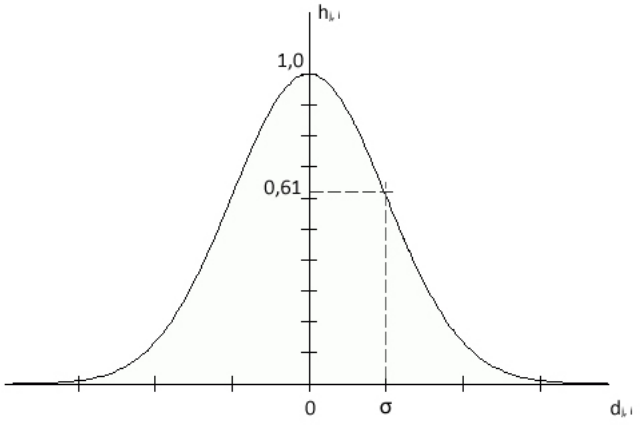

Figura 8: Função de vizinhança gaussiana

4. **Adaptação sináptica:** Nesta última fase os neurônios são capazes de reforçar a sua afinidade com o padrão de entrada responsável pela sua ativação, através de um aumento de seus valores individuais da função discriminante. Este efeito é alcançado através da aplicação de ajustes a seus pesos sinápticos, realizados de acordo com a fórmula de atualização (2.31) abaixo:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)h_{j,i(x)}(n)(\mathbf{x}(n) - \mathbf{w}_j(n)) \tag{2.31}$$

   Onde $\eta(n)$ é a taxa de aprendizagem e $h_{j,i(x)}(n)$ é a função de vizinhança centrada em torno do neurônio vencedor i($\mathbf{x}$), ambos variados dinamicamente durante a aprendizagem para a obtenção de melhores resultados.

5. **Continuação:** O algoritmo volta seguidamente para o passo 2 até que não mais se observe modificações significativas no mapa de características.

Além dos passos listados acima, o processo de treinamento da rede SOM pode ainda ser decomposto em dois estágios: a fase de auto-organização ou ordenação (*coarse-mapping*) e a fase de convergência (*fine-tuning*).

A primeira fase é responsável pela ordenação topológica dos vetores de peso e pode exigir até mais de 1000 iterações. Nela, a taxa de aprendizagem geralmente é inicializada com um valor próximo a 0,1, decrescendo gradualmente durante as iterações mas permanecendo acima de 0,01. A função de vizinhança, por sua vez, inclui inicialmente quase a totalidade dos neurônios da grade e diminui o raio de interação com o tempo.

Na fase de convergência, é realizada uma sintonia fina do mapa de características (por isso *fine-tuning*, no inglês), de maneira a produzir uma quantização estatística precisa do

espaço de entrada. Nela, o número de iterações deve ser muito superior ao da primeira fase, a taxa de aprendizagem deve ser mantida constante em um valor pequeno - geralmente 0,01 - e a função de vizinhança deve contar apenas com os vizinhos mais próximos so neurônio vencedor, podendo inclusive reduzir a um ou zero neurônios vizinhos.

2.2.2.3 Propriedades do Mapa de Características

Supondo que Φ represente uma transformação não-linear realizada pelo algoritmo SOM e chamada de mapa de características, que mapeia o espaço de entrada H para o espaço de saída A ($\Phi : H \rightarrow A$), criamos uma abstração para a equação (2.29), cuja finalidade é definir a localização de um neurônio vencedor $i(\mathbf{x})$ em resposta a um vetor de entrada $\mathbf{x}$. Tomando o contexto neurobiológico como fonte para comparação, o espaço de entrada H poderia representar o conjunto de coordenadas de receptores somestésicos densamente distribuídos sobre a superfície do corpo, enquanto A representaria o conjunto de neurônios localizados na camada do córtex cerebral à qual os receptores somestésicos estão confinados. O princípio é o mesmo para a rede neural artificial.

Esse mapa de características Φ, calculado pelo algoritmo SOM, apresenta algumas propriedades importantes:

1. **Aproximação do espaço de entrada:** O mapa de características Φ, representado pelo conjunto de vetores de pesos sinápticos $\mathbf{w}_j$ no espaço de saída A, fornece uma boa aproximação para o espaço de entrada H, tendo em mente que o objetivo básico do algoritmo SOM é armazenar um conjunto grande de vetores de entrada $\mathbf{x} \in H$, encontrando um conjunto menor de protótipos $\mathbf{w}_j \in A$. Essa característica resulta na compressão de dados, a partir de uma boa aproximação.

2. **Ordenação Topológica:** O mapa de características Φ calculado pelo algoritmo SOM é ordenado de modo topológico, no sentido de que a localização espacial de um neurônio na grade corresponde a um domínio particular ou característica dos padrões de entrada, em consequência direta da equação de atualização (2.31) que força o vetor peso sináptico $\mathbf{w}_i$ do neurônio vencedor $i(\mathbf{x})$ a se mover em direção ao vetor de entrada $\mathbf{x}$.

3. **Casamento de densidade:** O mapa de características Φ reflete variações na estatística da distribuição da entrada: regiões no espaço de entrada H de onde vetores de amostra $\mathbf{x}$ são retirados com uma alta probabilidade de ocorrência são mapeadas para domínios maiores do espaço de saída A, e portanto com melhor resolução que

regiões em H das quais vetores de amostra $\mathbf{x}$ são retirados com uma baixa probabilidade de ocorrência.

4. **Seleção de características:** Resultando de um casamento natural das características de 1 a 3, dizemos que, a partir de dados do espaço de entrada com uma distribuição não-linear, o mapa auto-organizável é capaz de selecionar um conjunto das melhores características para aproximar a distribuição adjacente.

2.2.2.4 Considerações Gerais

Qualquer tarefa de processamento com maior nível de dificuldade requer a organização da informação disponível e, em consequencia disso, muitos dos modelos que processam os valores de entrada sem estruturação exibem baixa convergência e pouca eficiência no processo de generalização. Segundo Kohonen (1990), para descrever o papel de um item no comportamento da série, é necessário que esse dado seja apresentado com uma quantidade suficiente de contexto e destaca ainda que, se for necessário trabalhar com sistemas mais extensos, pode ser mais eficiente o desenvolvimento de estruturas hierárquicas. Segundo ele, em um sistema mais natural, os módulos de uma estrutura hierárquica correspondem a áreas contíguas em um contexto mais amplo, onde cada uma dessas áreas recebe um tipo diferente de entrada externa, como acontece com as áreas do córtex cerebral.

O mapa auto-organizável captura características importantes contidas no espaço de entrada e os estrutura, fornecendo uma representação organizada dos dados. A definição de Haykin (2001) também serve de motivação para o desenvolvimento de uma arquitetura hirárquica como a descrita no capítulo 3. Segundo ele, um mapa auto-organizável é "um mapa topográfico dos padrões de entrada no qual as localizações espaciais (i.e. coordenadas) dos neurônios na grade são indicativas das características estatísticas intrínsecas contidas nos padrões de entrada".

Assim, o mapa da características formado pela rede SOM, fornecido pela sua camada de saída, pode se tornar uma fonte de informação importante sobre uma série temporal, incluindo, por exemplo, informações contextuais e estatísticas. A inserção dessas informações em um modelo com aprendizagem supervisionada possibilita um aumento em sua capacidade de generalização e na eficiência de sua previsão.

2.2.3 SVM - Máquinas de Vetor de Suporte

2.2.3.1 Princípios Básicos

As máquinas de vetor de suporte (SVM, do inglês *support vector machine*) são uma classe de máquinas de aprendizagem proposta por Vladimir Vapnik (CORTES; VAPNIK, 1995) que se utiliza de aprendizagem supervisionada e pode ser utilizada tanto para a classificação de padrões quanto para problemas de regressão linear. A sua derivação é baseada fortemente nos conceitos da teoria estatística da aprendizagem (VAPNIK, 1998), também conhecida como teoria VC (em homenagem a seus criadores Vapnik e Chervonenkis), que determina as propriedades necessárias para que uma máquina de aprendizagem seja capaz de generalizar bem para dados desconhecidos.

A idéia principal deste tipo de rede é construir um hiperplano como superfície de decisão de tal forma que a margem de separação entre os exemplos positivos e negativos seja máxima (HAYKIN, 2001), como ilustrado pela Figura 9, para o caso de um problema de classificação. Estabelecida sobre a teoria do método de minimização estrutural de risco, a SVM se mostra especialmente resistente também ao problema de *over-fitting*, atingindo frequentemente um "alto desempenho de generalização na solução de problemas de previsão em séries temporais" (CAO, 2002) e, também, na classificação de padrões.

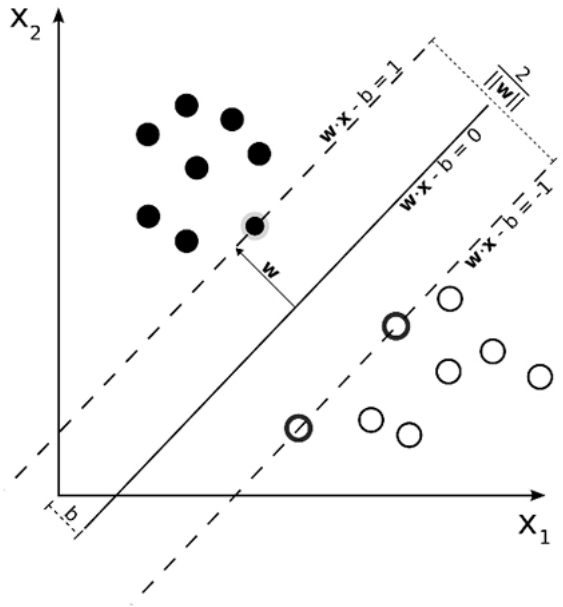

Figura 9: Hiperplano - SVM para classificação

O projeto da máquina depende diretamente da extração de um subconjunto dos dados de treinamento que representem características estáveis dos dados, chamados de *vetores de suporte*, através de um algoritmo. Para cada padrão de entrada é calculado um núcleo

do produto interno entre essa entrada e cada vetor de suporte, que funcionam como unidades ocultas da rede. Dependendo de como esse núcleo interno é gerado, podemos contruir diferentes máquinas de aprendizagem (polinomial, função de base radial), que se caracterizam por superfícies de decisão não-lineares próprias. Assim, podemos dizer que a máquina de vetor de suporte é uma rede alimentada adiante com uma única camada oculta de unidades não-lineares, formada pelos vetores de suporte extraídos pelo algoritmo. Tendo em mente que os vetores de suporte são apenas uma fração do espaço de entrada, é necessário enfatizar que o próprio algoritmo define a quantidade ótima de nós ocultos da rede, liberando o usuário da escolha deste parâmetro.

Seu treinamento equivale à solução de um problema de otimização quadrática, de maneira que a resposta encontrada pela máquina é sempre única e globalmente ótima, resolvendo um problema recorrente em outros tipos de redes neurais, como o perceptron de múltiplas camadas: a obtenção de falsas respostas devido aos mínimos locais da superfície de erro.

2.2.3.2 Máquinas de Vetor de Suporte para Regressão

Segundo Cortes e Vapnik (1995), a máquina de vetor de suporte conceitualmente implementa a seguinte idéia: vetores de entrada são mapeados não-linearmente para um espaço de características de alta dimensionalidade, onde uma superfície de decisão linear é construída. Propriedades especiais da superfície de decisão garantem alta habilidade de generalização para a máquina de aprendizagem.

Segundo Smola e Schölkopf (2004), na regressão ε-SV (comumente chamada de ε-SVR) desenvolvida por Vapnik (1995), o objetivo da máquina de suporte é encontrar uma função $f(x)$ que tenha no máximo um desvio ε dos valores desejados d_i para todos os dados de treinamento e, ao mesmo tempo, seja a mais suave possível. A rede executa essa regressão através da minimização do risco, onde o risco é dado pela função de perda insensível a ε de Vapnik (que será descrita adiante).

Considerando um conjunto de dados $G = \{(\mathbf{x}_i, d_i)\}_i^n$, onde $\mathbf{x}_i$ é o vetor de entrada, d_i é o valor desejado e n o total de padrões, a máquina de vetor de suporte aproxima a função utilizando a seguinte equação:

$$y = f(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x}) + b \tag{2.32}$$

Onde $\phi(\mathbf{x})$ é o espaço de características de alta dimensionalidade mapeado não-

linearmente a partir dos dados de entrada $\mathbf{x}$. Os coeficientes $\mathbf{w}$ e b são estimados através da minimização da função de custo 2.33:

$$R_{SVMs}(C) = C\frac{1}{n}\sum_{i=1}^{n} L_{\varepsilon}(d_i, y_i) + \frac{1}{2}||\mathbf{w}||^2 \tag{2.33}$$

Sendo $L_{\varepsilon}(d_i, y_i)$ a função de perda dada por:

$$L_{\varepsilon}(d, y) = \begin{cases} |d-y| - \varepsilon \;\; se\ |d-y| \geq \varepsilon \\ 0 \quad caso\ contrario \end{cases} \tag{2.34}$$

A função de perda 2.34 dá ao modelo a vantagem da escolha de pontos de dados esparsos para a representação da função de decisão, resultando na escolha dos vetores de suporte apenas quando o seu valor torna-se diferente de 0.

O segundo termo da equação (2.33), dado por $\frac{1}{2}||\mathbf{w}||^2$, é o termo de regularização, de maneira que o parâmetro C torna-se responsável pelo balanceamento entre os termos que definem o risco empírico (a função de perda) e a regularização (TAY; CAO, 2001a). Um aumento em C resulta no aumento da importância do risco empírico em relação ao parâmetro de regularização, ou seja, o parâmetro C determina um balanceamento entre a suavidade da curva e a quantidade de desvios maiores que ε tolerados pelo modelo. Tanto C quanto ε são parâmetros definidos empiricamente pelo usuário do modelo, de acordo com a sua necessidade. O desenvolvimento de uma abordagem consistente para a seleção desses parâmetros ainda é um campo de pesquisa em aberto.

Para obter as estimativas de $\mathbf{w}$ e b, a equação (2.33), a ser minimizada, é transformada para a função primitiva dada pela equação (2.35) através da introdução de *variáveis soltas* positivas ξ_i e ξ_i^*, como segue:

$$R_{SVMs}(\mathbf{w}, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2.35}$$

que obedece às seguintes restrições:

$$\begin{cases} d_i - \mathbf{w}\phi(\mathbf{x}_i) - b_i \leq \varepsilon + \xi_i \\ \mathbf{w}\phi(\mathbf{x}_i) + b_i - d_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{2.36}$$

A formulação acima corresponde à utilização da anteriormente citada função de perda

insensível a ε de Vapnik, descrita por:

$$|\xi|_\varepsilon := \begin{cases} 0 \ se \ |\xi| \leq \varepsilon \\ |\xi - \varepsilon| \ caso \ contrario \end{cases} \tag{2.37}$$

A Figura 10 ilustra a situação graficamente. De acordo com a função de custo 2.37, somente os pontos que se encontram fora da área sombreada contribuem para o custo e são escolhidos como vetores de suporte.

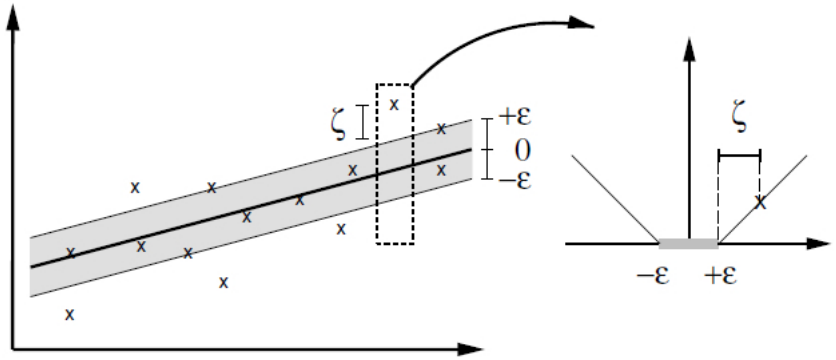

Figura 10: Hiperplano - SVM para regressão (SMOLA; SCHöLKOPF, 2004)

O problema de otimização apresentado em 2.35 pode ser resolvido mais facilmente em sua formulação dual, que estende a SVM para funções não-lineares através de um método padrão de dualização baseado em multiplicadores de Lagrange. Assim, a equação (2.32), que representa a solução da máquina de vetor de suporte, toma a seguinte forma (VAPNIK, 1995):

$$f(\mathbf{x}, a_i, a_i^*) = \sum_{i=1}^{n} (a_i - a_i^*) K(\mathbf{x}, \mathbf{x}_i) + b \tag{2.38}$$

Onde a_i e a_i^* são os multiplicadores de Lagrange, obtidos através da maximização da função dual de 2.35, e que obedecem as seguintes restrições:

$$\begin{cases} \sum_{i=1}^{n} (a_i - a_i^*) = 0 \\ a_i, a_i^* \in [0, C] \end{cases} \tag{2.39}$$

O termo $K(\mathbf{x}_i, \mathbf{x}_j)$ da equação (2.38) acima é definido como a **função de kernel**. O valor do kernel é calculado pelo produto interno entre dois vetores $\mathbf{x}_i$ e $\mathbf{x}_j$ no espaço de características, de maneira que $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) * \phi(\mathbf{x}_j)$. A vantagem da utilização da função de kernel reside no fato de que, através dela, é possível manipular um espaço de

características de dimensionalidade arbitrária sem a necessidade de se computar explicitamente o mapa $\phi(\mathbf{x})$. Segundo Vapnik (1995), qualquer função que satisfaça as condições do Teorema de Mercer (MERCER, 1909) pode ser utilizada como função de kernel. Dois exemplos típicos de funções de kernel (HAYKIN, 2001) são o kernel polinomial, definido por:

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i \mathbf{x}_j) + 1)^p \tag{2.40}$$

e o kernel de função de base radial, definido por:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||^2\right) \tag{2.41}$$

Onde os parâmetros p e σ^2 equivalem, respectivamente, ao grau do polinônio e σ^2 a largura do kernel gaussiano, escolhidos pelo usuário.

A equação (2.38) é a expansão dos vetores de suporte, onde $\mathbf{w}$ pode ser completamente descrito como uma combinação linear dos padrões de entrada $\mathbf{x}_i$. Somente uma parcela dos coeficientes $(a_i - a_i^*)$ assume valores diferentes de zero (TAY; CAO, 2001a). Os pontos associados a eles possuem erros de aproximação iguais ou superiores a ε e são escolhidos como vetores de suporte. Nesse sentido, a complexidade do modelo torna-se independente da dimensionalidade do espaço de entrada, dependendo apenas da quantidade de vetores de suporte. Além disso, torna-se desnecessária a computação explícita do parâmetro $\mathbf{w}$, uma vez que o algoritmo consiste na realização de um produto entre vetores de dados. A escolha da função de kernel deve ser realizada cuidadosamente, pois é ela que implicitamente define a estrutura do espaço de características de alta dimensionalidade $\phi(\mathbf{x})$ e controla a complexidade da solução final.

Geralmente, quanto maior o valor de ε, menor o número de vetores de suporte e mais esparsa a representação da solução. No entanto, é preciso salientar que valores grandes de ε diminuem a precisão da aproximação e geram resultados menos próximos do esperado.

Finalmente, podemos dizer que, do ponto de vista da implementação, o treinamento das máquinas de vetor de suporte equivale à solução de um problema de programação quadrática. Entre os algoritmos para a solução do problema de otimização destaca-se o proposto por Smola e Schölkopf (2004), específico para máquinas de vetor de suporte para regressão.

2.2.3.3 Considerações Gerais

As máquinas de vetor de suporte diferem dos perceptrons de múltiplas camadas de maneira fundamental. Na abordagem mais convencional, utilizada pela rede MLP, a complexidade do modelo é controlada mantendo-se o número de características, ou de neurônios ocultos, baixo. A máquina de vetor de suporte, por sua vez, controla a complexidade do modelo de forma independente da dimensionalidade do espaço de características, que é feita propositalmente grande para possibilitar a construção de uma superfície de decisão na forma de um hiperplano naquele espaço. O bom desempenho de generalização é controlado pela imposição de certas restrições sobre a construção do hiperplano de separação, resultando na extração de uma fração dos dados de treinamento como vetores de suporte.

A maldição da dimensionalidade é evitada através da utilização da noção de um núcleo de produto interno e resolução da forma dual do problema de otimização restrito formulado no espaço de entrada. Uma importante razão para a utilização da formulação dual é, exatamente, evitar a necessidade de definir e calcular os parâmetros do hiperplano ótimo em um espaço de dados possivelmente de alta dimensionalidade. Desta maneira, o treinamento da SVM torna-se, geralmente, um problema de programação quadrática, garantindo que se encontre um extremo global da superfície de erro, onde o erro se refere à diferença entre a resposta desejada e a saída da SVM, e que a computação seja realizada eficientemente.

As máquinas de vetor de suporte são um raro exemplo de metodologia onde intuição geométrica, matemática elegante, garantias teóricas e algoritmos práticos se encontram. Seus resultados finais são estáveis, reproduzíveis e independentes do algoritmo específico utilizado para otimizar o modelo. Se dois usuários utilizarem o mesmo modelo, mesmos parâmetros e dados, terão os mesmos resultados (BENNETT; CAMPBELL, 2000).

No entanto, os resultados obtidos pela SVM são de difícil interpretação, devido à quantidade limitada de informação interpretável apresentada pelos vetores de suporte definidos pelo algoritmo. Além disso, pode-se ainda destacar que, em termos de tempo de execução, as máquinas de vetor de suporte ainda são, atualmente, mais lentas que as outras redes neurais - como a MLP - para uma desempenho similar de generalização.

2.3 Trabalhos Existentes

A previsão em séries temporais financeiras sempre foi considerada uma tarefa com alto nível de dificuldade, e tem sido objeto de estudo de muitos pesquisadores, cuja linha de trabalho e ferramentas de previsão variam em uma ampla coleção de modelos e arquiteturas. Muitos deles chegam a considerar mesmo que se trata de uma tarefa próxima do impossível e que esforços nessa área configuram-se quase sempre como perda de tempo e recursos.

Esta questão é explorada por Clements, Franses e Swanson (2004) em seu artigo, que discute exatamente o estado da arte no campo da estimação, avaliação e seleção de modelos de previsão para séries financeiras e econômicas. O artigo revisa questões teóricas e resultados empíricos obtidos historicamente chegando à conclusão de que, apesar de algumas evidências contrárias, a construção de modelos de previsão para este tipo de série possui razões para otimismo. Os autores nos sugerem que, uma vez que nosso mundo é intrinsecamente não-linear, apenas modelos que explorem esta característica - como redes neurais, algoritmos genéticos, etc - possuem um horizonte promissor. Além disso, o artigo destaca que muito trabalho ainda está por ser realizado e que o aumento da capacidade computacional no decorrer dos anos só tende a tornar ainda melhores os resultados obtidos pelos modelos, que devem evoluir e se tornar cada vez mais detalhados e complexos.

As ponderações de Clements, Franses e Swanson (2004) se confirmam na tendência atual da área de previsão em séries financeiras. Ao realizar uma consulta em um mecanismo de busca de artigos da internet (GOOGLE..., 2010) com os termos *"forecasting financial time series"* ("prevendo em séries temporais financeiras", em português), obtém-se seis resultados envolvendo redes neurais artificiais entre os dez mais relevantes, sendo que quatro deles destacam-se ainda por apresentar modelos baseados em máquinas de vetor de suporte, um dos focos desta dissertação.

Pérez-Cruz e Bousquet (2004) destacam em seu trabalho que muitos problemas surgidos no processamento digital de sinais são de natureza estatística e necessitam de métodos de análise de dados automáticos. Os algoritmos utilizados nestas tarefas são geralmente lineares e sua transformação para o processamento não-linear é frequentemente nebulosa. Pesquisadores da área podem, portanto, beneficiar-se de um conhecimento mais profundo sobre métodos de *kernel* (KM, de *kernel methods*) e máquinas de vetor de suporte (SVM de *support vector machines*), pois estas técnicas apresentam uma maneira diferente - e mais clara - de se levar em consideração as não-linearidades inerentes ao problema sem

perda das propriedades originais dos métodos lineares.

Kim (2003) apresenta um exemplo de aplicação das máquinas de vetor de suporte para previsão de índices financeiros relacionados ao mercado de ações. O artigo estuda a viabilidade desta técnica na área financeira através de uma comparação com uma rede neural MLP e com a técnica de *case-based reasoning* (CBR). Os resultados obtidos apontam que a SVM se apresenta como uma alternativa promissora para a previsão no mercado de ações, alcançando resultados melhores que as duas técnicas utilizadas como parâmetros de comparação.

Cao e Tay (2001) exploram o assunto de forma semelhante e comparam a eficiência do modelo SVM com a rede MLP utilizando dados provenientes do *Chicago Mercantile Market*. Neste trabalho várias métricas são utilizadas para representação do erro e, mais uma vez, a máquina de vetor de suporte se sobressai em relação ao modelo MLP. Os autores concluem que a SVM é vantajosa devido a características de sua construção como a utilização da minimização estrutural do risco, o menor número de parâmetros livres e o cálculo de solução global. Os autores destacam ainda a escolha apropriada dos parâmetros livres da SVM, e sua importância é demonstrada através da variação da eficiência nos testes.

Cao e Tay (2003) voltam a discutir o assunto dos parâmetros livres em outro artigo, onde é proposto um modelo onde estes parâmetros sejam adaptativos e não fixos, de maneira a incorporar as mudanças de regime da série financeira e atingir melhores resultados de generalização (ajuste do parâmetro C) e menor número de vetores de suporte (parâmetro ε). Os resultados obtidos pela SVM são comparados aos da MLP e de redes de função de base radial (RBF, de *radial-basis function*) e, novamente, a SVM se sobressai em relação à MLP mas, neste caso, não apresenta melhorias significativas em relação à rede RBF.

Entretanto, ao comparar os modelos SVM e RBF, Müller et al. (1997) nos mostram que os resultados da SVM alcançam uma melhoria da ordem de até 30% em relação aos obtidos pela rede RBF em uma série teórica chamada *Makey-Glass* (MAKEY; GLASS, 1977) e em um conjunto de dados padrão chamado *Santa Fe Competition - Set D* (WEIGEND; GERSHENFELD, 1994). No artigo os autores ajustam os parâmetros empiricamente e observam ainda que máquinas de vetor de suporte são especialmente eficientes ao trabalhar com dados esparsos (poucos dados em uma alta dimensionalidade).

A preocupação com os parâmetros livres do modelo é também compartilhada por Kaastra e Boyd (1996). O artigo apresenta um procedimento para a condução de experi-

mentos envolvendo redes neurais e séries temporais financeiras e mostra grande interesse na seleção dos parâmetros livres. Segundo os autores, a execução deste tipo de tarefa muitas vezes é comprometida pela seleção de parâmetros, que ocorre quase sempre na base da tentativa e erro.

Yang, Chan e King (2002) também propõem a utilização de máquinas de vetor de suporte para regressão (SVR, de *support vector regression*) em séries temporais financeiras. Os autores percebem que dados financeiros são frequentemente ruidosos e possuem volatilidade variável no tempo e que, além disso, a medida de desvio padrão apresenta-se como uma boa medida de volatilidade para a série. Partindo deste pressuposto, são realizadas modificações no algoritmo da LIBSVM (CHANG; LIN, 2001) de forma que mudanças na volatilidade da série se reflitam em variações na margem da SVR. Os resultados obtidos mostram que a utilização da medida de desvio padrão para cálculo dinâmico de uma margem variável (ou melhor, adaptativa) gera bons resultados em relação ao modelo padrão, de margem fixa e simétrica.

A utilização de mapas auto-organízáveis na área financeira também tem sido cada vez mais acentuada. Deboeck (1998) nos diz que a mineração de conhecimento é a extração não-trivial de conhecimento implícito, previamente desconhecido e potencialmente útil da fonte de dados disponível. O autor apresenta as redes SOM como "descobridoras" de conhecimento e destaca que os resultados obtidos podem ser melhorados através de sua utilização em conjunto com técnicas como lógica *fuzzy*, algoritmos genéticos e outros modelos neurais. O exemplo utilizado para ilustração é a seleção de fundos de investimento semelhantes e sua classificação em sub-grupos e os resultados obtidos são muito bons.

A inclusão dos mapas auto-organizáveis no panorama da análise de séries temporais financeiras proporciona uma variedade de possibilidades na construção de modelos, especialmente envolvendo mais de uma arquitetura de redes neurais. Os diversos modelos desenvolvidos quase sempre se utilizam de um pré-processamento dos dados realizado em seu primeiro nível através de uma rede SOM, seguido de um estágio supervisionado responsável pela previsão - geralmente composto por uma rede MLP ou uma máquina de vetor de suporte. No entanto, muitas vezes os modelos se diferenciam também pela função do mapa auto-organizável na estrutura do previsor.

Hsu et al. (2009) destacam que o relacionamento entre a série de preços de ações e os modelos de previsão é muito dinâmico e que este tipo de série temporal possui uma natureza não-estacionária, ignorada por parte dos pesquisadores. Os autores nos mostram que o emprego de uma técnica individual de previsão não consegue modelar efetivamente

esta característica particular da série financeira e que a solução potencial é formar modelos híbridos, com diferentes técnicas artificiais. Desta maneira, é desenvolvido no decorrer do artigo uma arquitetura de dois estágios formada em seu primeiro nível por um mapa auto-organizável (SOM) cuja função é decompor o espaço de entrada em regiões onde os dados possuam distribuições estatísticas similares e capturar a não-estacionariedade da série, seguido por máquinas de vetor de suporte para regressão (SVR) especialistas utilizadas para prever os índices financeiros para cada um dos grupos formados pela rede SOM. O modelo é testado com sete séries de mercados financeiros importantes e os resultados apresentados são melhores que os da SVM pura em todos os testes, ainda que o ganho não seja particularmente substancial.

Tay e Cao (2001b) também utilizam este tipo de abordagem na solução do problema de previsão nas séries financeiras. Em seu artigo é desenvolvido um modelo semelhante ao apresentado por Hsu et al. (2009), mas apresenta agora um critério para a divisão em sub-grupos. O artigo mostra uma arquitetura estruturada em árvore cujo objetivo é evitar que o número de regiões seja pré-definido. No primeiro estágio, redes SOM dividem o conjunto de dados em dois sub-grupos repetidamente, até que um critério seja atingido (número mínimo de padrões por sub-grupo, por exemplo). Em outras palavras, não é somente o grupo inicial que sofre a divisão, mas também os sub-grupos - daí a estrutura de árvore. No segundo estágio, as máquinas de vetor de suporte especialistas são ajustadas com a melhor função de *kernel* e os melhores parâmetros livres para cada um dos sub-grupos formados no primeiro estágio. O modelo alcança uma performance significativamente melhor em relação a SVM pura, além de convergir mais rapidamente para a solução. Cao (2003) volta a trabalhar com a mesma arquitetura em problemas de outras áreas, apresentando novamente bons resultados.

A arquitetura de dois estágios apresentada por esses autores, onde o primeiro estágio, formado por um mapa auto-organizável, é responsável pela segmentação dos dados, e o segundo, formado por máquinas de vetor de suporte especialistas, responde pela tarefa da previsão em si, tornou-se popular e foi também empregada com sucesso em outras áreas como, por exemplo, a previsão em curto prazo do preço da eletricidade (FAN; MAO; CHEN, 2007) e a previsão de carga em sistemas elétricos (FAN; CHEN, 2006), onde a mudança de regime também se apresenta como uma característica fundamental.

Armano, Marchesi e Murru (2005) utilizam um modelo híbrido semelhante para abordar séries financeiras, mas no papel normalmente realizado pelo mapa auto-organizável é utilizada a técnica de algoritmos genéticos. O autor explica que a previsão utilizando um

modelo neural isolado é muito difícil devido à mudança de regime da série, de tempos em tempos. Características como a volatilidade se apresentam em forma de *clusters* e mudam no decorrer do tempo, muitas vezes em ciclos. Assim, o modelo desenvolvido tem como objetivo identificar modelos locais para segmentos específicos da série em uma abordagem referida como "baseada em contexto" e formar um grupo de redes neurais especialistas (MLP) onde o responsável pela previsão no contexto atual é definido através dos algoritmos genéticos. Os resultados obtidos pelo modelo híbrido são superiores aos da rede neural simples.

Apresentando um modelo semelhante aos já citados mas divergindo no papel dos mapas auto-organizáveis no modelo hierárquico, Carpinteiro et al. (2007) introduzem uma arquitetura formada por duas redes SOM - uma sobre a outra - e um perceptron de camada única (SLP, de *single layer perceptron*). Neste modelo o papel da SOM não é segmentar a série em *clusters* de características estatísticas similares, mas processar de maneira eficiente a informação de contexto presente na série histórica de previsão de carga elétrica, formando uma memória significativa de eventos passados e possibilitando melhores resultados de previsão. Os resultados obtidos para previsão de carga elétrica a longo prazo foram muito bons e superiores aos obtidos pelo MLP, utilizado como parâmetro de comparação.

A mesma arquitetura - baseada na construção de contexto para os dados de entrada - já havia apresentado resultados interessantes em outra área de pesquisa de redes neurais artificiais, obtendo resultados significativos em classificação no universo de análise de sinais musicais (CARPINTEIRO; BARROW, 1996). Arquitetura semelhante será o objeto de estudo desta dissertação e o modelo será descrito no capítulo 3.

3 Modelo Neural Hierárquico

3.1 Motivação

Os modelos neurais hierárquicos possuem memórias mais extensas do que modelos mais simples em relação a eventos passados e, por essa razão, podem ser utilizados com sucesso na análise e previsão de séries temporais. Sua característica de memória mais extensa deve-se à segmentação da série temporal realizada pelo modelo: Modelos hierárquicos segmentam a série temporal em grupos de elementos básicos, que juntos representam diferentes contextos para os dados. É essa segmentação que facilita o processo de aprendizagem e torna possível a realização de previsões de boa qualidade. Hawkins e Blakeslee (2004) atribuem a própria inteligência humana ao mecanismo utilizado pelo cérebro que, também baseado em memória e contexto, realiza previsões de eventos futuros continuamente.

Os problemas relacionados ao domínio visual apresentam-se como uma boa analogia e demonstram o conceito que desejamos esclarecer. Segundo Anderson (1990), o reconhecimento de um padrão visual complexo envolve a análise de características, sendo que esta análise consiste na divisão do padrão complexo em um conjunto de características primitivas, seguida do reconhecimento de cada uma destas características e, finalmente, no reconhecimento da combinação dessas características para identificação do padrão como um todo.

O mesmo procedimento pode ser utilizado em problemas no domínio musical. Segundo Drake e Palmer (1993) e Lerdahl e Jackendoff (1983), o reconhecimento de padrões musicais também consiste na divisão do padrão em um conjunto de fragmentos musicais e no reconhecimento de cada um destes fragmentos individualmente para, finalmente, reconhecer a combinação dos fragmentos. Basicamente, o conceito resume-se em tomar uma tarefa de grande complexidade e dividi-la em diversas tarefas menores e menos complexas.

Em modelos neurais hierárquicos, a análise de uma série temporal complexa é reali-

zada de forma semelhante à dos padrões visuais e musicais. Para realizar a previsão, o córtex cerebral necessita de uma maneira para memorizar e armazenar conhecimento sobre uma sequência de eventos. Para prever novos eventos, o córtex precisa construir representações invariantes, ou generalizar a partir destes dados. Assim, pode-se dizer que o cérebro precisa criar e armazenar um modelo do mundo como ele realmente é, independentemente de como ele se apresenta em diferentes circunstâncias (HAWKINS; BLAKESLEE, 2004). O mesmo pode ser dito sobre o modelo hierárquico desenvolvido durante este trabalho, onde uma representação do problema é formada, armazenada nas sinapses da rede e disponibilizada para consulta.

As arquiteturas hierárquicas são formadas por dois ou mais modelos neurais, dispostos acima um do outro. O modelo diposto na parte inicial ou inferior torna-se, assim, responsável pela segmentação da série temporal em uma série de características primitivas, sua codificação em contextos e, também, pela análise individual de cada um destes contextos. O modelo disposto na parte final ou superior, por sua vez, é responsável pela interpretação da combinação destas características, análise do contexto construído e apresentação do valor final da previsão.

3.2 Componentes Estruturais

O relacionamento entre uma série financeira e os modelos de previsão é muito dinâmico. Esse tipo de série apresenta características, como a volatilidade, que se apresentam na forma de *clusters*, mudando de regime no decorrer do tempo. Por isso, o emprego de uma técnica individual de previsão não consegue modelar efetivamente suas características particulares e a solução para este problema é a utilização de modelos híbridos - com mais de um tipo de rede neural (HSU et al., 2009).

O modelo neural hierárquico (HNM, do inglês *Hierarchical neural model*) empregado nos experimentos descritos no capítulo seguinte (Capítulo 4) e desenvolvido no decorrer deste trabalho é formado por dois modelos neurais: um mapa auto-organizável (SOM, de *self-organizing map*) (KOHONEN, 2001) e uma máquina de vetor de suporte (SVM, de *support vector machine*) (CORTES; VAPNIK, 1995), disposta sobre este mapa. O modelo formado está representado de maneira simplificada na Figura 11, onde Λ representa uma função de transferência que será explicada adiante. No modelo, o conhecimento é representado pelo próprio estado de ativação da rede, distribuído pela sua estrutura.

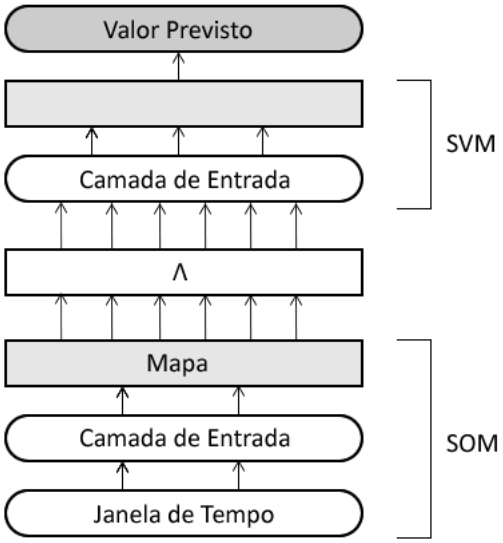

Figura 11: Modelo Neural Hierárquico (HNM)

Para começar, podemos dizer que a mineração de conhecimento é a extração não-trivial de conhecimento implícito, previamente desconhecido e potencialmente útil da fonte de dados disponível. As redes SOM são "mineradoras" de conhecimento e seus resultados podem ser melhorados através de sua utilização em conjunto com outras técnicas como redes neurais (DEBOECK, 1998).

Colocando de outra forma, o mapa auto-organizável é capaz de selecionar um conjunto das melhores características do conjunto de dados que representa o problema. O mapa de características formado fornece uma boa aproximação para o espaço de entrada e armazena um grande conjunto de dados em um conjunto menor de vetores de peso, gerando uma representação reduzida dos dados. Além disso, o mapa de características reflete variações estatísticas da distribuição de entrada: regiões do espaço de entrada que possuam maior probabilidade de ocorrência são mapeadas para regiões maiores do espaço de características. Desta maneira, cada parte da informação recebida pelo sistema é mantida em seu próprio contexto e neurônios especializados em cada uma delas podem interagir entre si através de conexões sinápticas curtas.

Portanto, o primeiro passo do modelo, representado pelo mapa auto-organizável, descreve o papel das entradas no comportamento da série temporal, organizando os dados e apresentando em seu espaço de características uma quantidade suficiente de informações sobre contexto e características estatísticas intrínsecas do conjunto de treinamento. Como a entrada do modelo é composta por valores passados da série dipostos em uma janela temporal, o mapa funciona também como a memória do modelo. A apresentação deste mapa de características como entrada para um modelo com aprendizagem supervisiona-

da possibilita um aumento na sua capacidade de generalização e, consequentemente, na eficiência da previsão.

As máquinas de vetor de suporte tornam-se candidatas ideais para o segundo passo do modelo, devido às suas características fundamentais: sua estrutura garante que se encontre um resultado global, através da solução de um problema de programação quadrática. Este tipo de máquina de aprendizagem combina essa garantia teórica com intuição geométrica, desenvolvimento matemático robusto e algoritmos práticos, formando um sistema singular. Seus resultados finais são estáveis, reproduzíveis e independentes do algoritmo específico utilizado em sua solução. Sua utilização tem sido muito divulgada no tratamento de problemas que envolvam o mercado financeiro e tem alcançado resultados quase sempre mais promissores que outros modelos como MLP e redes de função de base radial.

Em resumo, o modelo HNM é formado por um mapa auto-organizável e uma máquina de vetor de suporte, onde o papel do primeiro é processar de maneira eficiente a informação de contexto presente na série histórica e formar uma memória significativa de eventos passados, possibilitando melhores resultados na previsão desempenhada pela máquina de vetor de suporte.

3.3 Funcionamento do Modelo

Na estrutura formada, a rede SOM é responsável pela construção de contextos para a informação de entrada. Quando uma janela temporal é apresentada à sua camada de entrada em conjunto com informação referente à volatilidade da série, o modelo constrói contextos para aquelas janelas temporais, utilizando toda a extensão do mapa.

Existem duas vantagens relacionadas a este tipo de abordagem. Primeiramente, através dela torna-se desnecessária a preocupação com a codificação manual e antecipada de um contexto para os dados, uma vez que esta responsabilidade fica a cargo do modelo SOM, que os representa na extensão de seu mapa. Segundo, evita-se a representação de todos os contextos possíveis dados pelas muitas combinações possíveis de vetores de entrada, uma vez que a rede SOM somente contruirá os contextos necessários à aplicação e característicos ao tipo de dado que serve como entrada: uma série financeira, no caso desta dissertação.

Após a criação dos contextos, a SVM recebe a informação transmitida pelo mapa auto-organizável. Os dados recebidos por sua camada de entrada são agora constituídos não somente pela janela temporal ou valores explícitos da série estudada, mas sim pelas

informações referentes ao contexto histórico e estatístico no qual esta janela está inserida. É através destes dados que a máquina de vetor de suporte será capaz de nos apresentar uma previsão mais confiável.

O treinamento da SOM acontece em duas fases, chamadas de "ordenação" (*coarse-mapping*) e "convergência" (*fine-tuning*). Na fase de ordenação, a taxa de aprendizagem e o raio da vizinhança são reduzidos linearmente, enquanto na fase de convergência, os parâmetros são mantidos constantes. Os pesos iniciais são atribuídos aleatoriamente.

A informação construída pelo mapa da SOM é transmitida para a entrada da SVM através de uma função de transferência Λ, calculada para cada neurônio i de sua camada de saída. Dois tipos de função de transferência foram testados: A função gaussiana e uma função discreta. A função de transferência Λ_1 gaussiana foi definida de acordo com a equação (3.1), onde k é uma constante, σ é o raio da gaussiana e $\Psi(i,t)$ é a distância euclidiana entre o vetor $\mathbf{x}(t)$, dado pela entrada do modelo no tempo t, e o vetor peso $\mathbf{w}_i$ da unidade neural i.

$$\Lambda_1(\Psi(i,t)) = e^{-\frac{k[\Psi(i,t)]^2}{\sigma^2}} \tag{3.1}$$

A função de transferência discreta Λ_2, de aspecto retangular, foi definida como descrito na equação (3.2), sendo k uma constante escolhida pelo usuário, $N^*(t)$ a vizinhança (do inglês *neighbourhood*) da unidade vencedora i^*, e $\Phi(i, i^*(t))$ a distância no mapa entre a unidade i e a unidade vencedora $i^*(t)$.

$$\Lambda_2(\Psi(i,i^*(t))) = \begin{cases} 1 - k\Phi(i,i^*(t)) \; se \; i \in N^*(t) \\ 0 \; se \; i \notin N^*(t) \end{cases} \tag{3.2}$$

A distância $\Phi(i', i'')$ entre duas unidades i' e i'' quaisquer no mapa é calculada de acordo com a norma máxima (Equação 3.3), onde (l', c') e (l'', c'') são coordenadas das unidades i' e i'', respectivamente, no mapa.

$$\Phi(i', i'') = max\{|l' - l''|, |c' - c''|\} \tag{3.3}$$

A Figura 12 ilustra a diferença entre os dois tipos de função de transferência. Enquanto na função gaussiana o nível de ativação decresce continuamente com o aumento da distância em relação ao neurônio vencedor (círculo central), na função discreta os níveis de ativação decrescem em forma de degraus, impondo uma aparência retangular de

quantização à sua representação.

Figura 12: Funções de transferência: Gaussiana e discreta

Entretanto, independentemente da função de transferência utilizada, o funcionamento básico do sistema pode ser resumido como segue:

- Padrões de entrada, formados por valores recentes da série financeira, são apresentados à entrada da SOM com n neurônios na camada de saída. Caso o usuário julgue necessário, informações adicionais sobre o momento atual da série - como volatilidade - podem ser adicionadas à entrada;
- Uma função de transferência - discreta ou gaussiana - é aplicada, gerando n saídas no mapa de características da SOM. A escolha da função é realizada empiricamente, baseadas em seus resultados para cada problema em particular;
- As n saídas do mapa são apresentadas aos nós sensoriais da maquina de vetor de suporte, que realiza o processamento e gera uma resposta para a previsão.

Para a definição do melhor modelo a ser utilizado no decorrer dos experimentos, foram realizados muitos testes. A máquina de vetor de suporte foi testada com as quatro funções de *kernel* disponíveis na biblioteca de software LIBSVM (CHANG; LIN, 2001): Função de base radial (gaussiana), linear, polinomial e sigmóide. A rede SOM foi treinada com mapas de duas dimensões, variando desde 5x5 até 100x100 unidades, e funções de tranferência discreta e gaussiana com diversos valores de raio.

Os melhores resultados foram obtidos pelo modelo hierárquico HNM composto de uma rede SOM com mapa de 50x50 unidades utilizando a função de transferência discreta Λ_2 com valor de distância Φ igual a cinco, e uma SVM com função de kernel de base radial. Como a dinâmica de uma série financeira é fortemente não-linear, é intuitiva a constatação de que a utilização de um kernel não-linear, como a função de base radial, também tenha levado o sistema a resultados mais expressivos.

4 Experimentos e Resultados

4.1 Série Utilizada

O modelo de previsão desenvolvido no decorrer de todo o período de pesquisa tem como objetivo realizar a previsão de uma série temporal financeira, com uma margem de erro satisfatoriamente pequena, levando-se em consideração a inerente complexidade deste tipo de série e adotando como parâmetros de comparação alguns modelos já consagrados e frequentemente utilizados para a mesma finalidade.

Tendo em vista o objetivo proposto, foi escolhida como objeto de estudo a série temporal com o valor diário das cotas do Fundo de Investimento Banco do Brasil IBrX Indexado (FUNDOS..., 2010), disponibilizada no intervalo de tempo de aproximadamente sete anos e meio (2 de julho de 2002 a 31 de dezembro de 2009), e que conta com todas as características e dificuldades básicas de uma série financeira tradicional.

Este fundo, administrado pela BB Gestão de Recursos - Distribuidora de Títulos e Valores Mobiliários S.A., tem como objetivo aplicar recursos do investidor em cotas de fundos de investimento que apresentem uma carteira de ativos que reflita o comportamento da carteira teórica do IBrX - Índice Brasil.

O IBrX - Índice Brasil é um índice de preços que mede o retorno de uma carteira teórica composta por ações de 100 companhias abertas, selecionadas entre as mais negociadas na Bolsa de Valores de São Paulo (BOVESPA), no que diz respeito ao número de negócios e volume financeiro. Essas ações são ponderadas na carteira do índice pelo seu respectivo número de ações disponíveis para negociação no mercado.

O fundo aplica os recursos dos investidores em cotas de fundos de investimentos que possuam em suas carteiras ações e bônus de subscrição de ações negociados em bolsas de valores ou mercado de balcão organizado, títulos públicos federais, operações compromissadas lastreadas nesses títulos e cotas de fundos de investimento, obedecidos os limites estabelecidos em seu regulamento (BB DTVM, 2010). O fundo pode atuar no mercado de

derivativos para proteger parte de seu patrimônio ou para reproduzir uma posição em ações com a parcela de sua carteira que estiver direcionada para ativos de renda fixa, sendo vedada a exposição, a esses mercados, superior ao seu patrimônio líquido.

A Figura 13 mostra os setores das empresas cujas ações compunham a carteira de ações do fundo na data de 30 de junho de 2010:

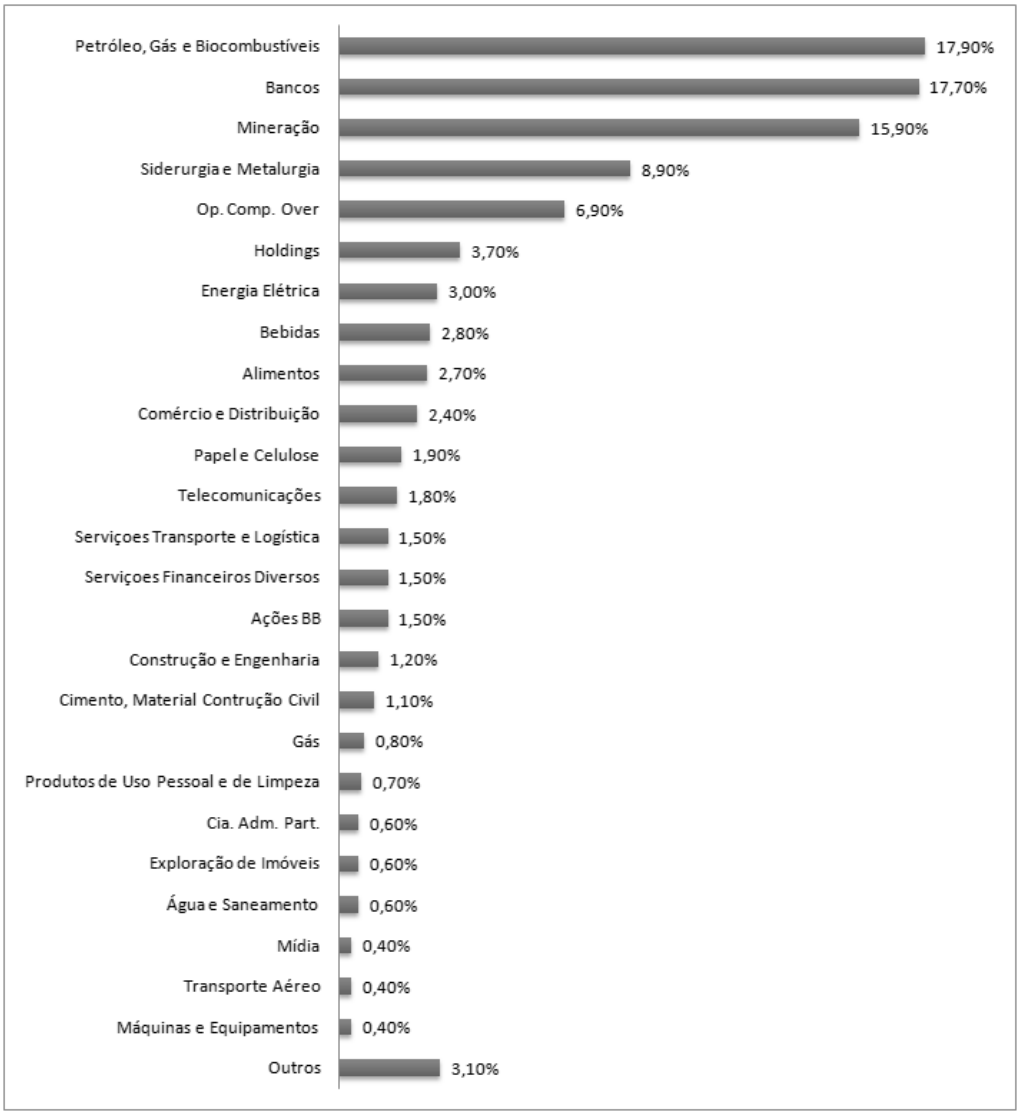

Figura 13: Composição da carteira do fundo de investimento IBrX (Jun/2010)

Portanto, em resumo, o fundo destina-se a investidores - clientes do Banco do Brasil S/A - que desejam aplicar seus recursos no mercado acionário, em troca de rentabilidade que acompanhe a variação do IBrX, e estejam dispostos a assumir os riscos inerentes a este mercado.

A Figura 14, a seguir, mostra graficamente o comportamento da série de preços das cotas do fundo de investimento no decorrer do tempo. Nela, é possível observar que a

série estudada é uma série não-estacionária, com grande intervalo de amplitude, tendência de crescimento, e com períodos razoavelmente bem definidos onde pode-se notar a estabilidade ou instabilidade do mercado. O processamento realizado na série para tornar possível a previsão e tornar explícitas suas características mais importantes será descrito na seção 4.4.

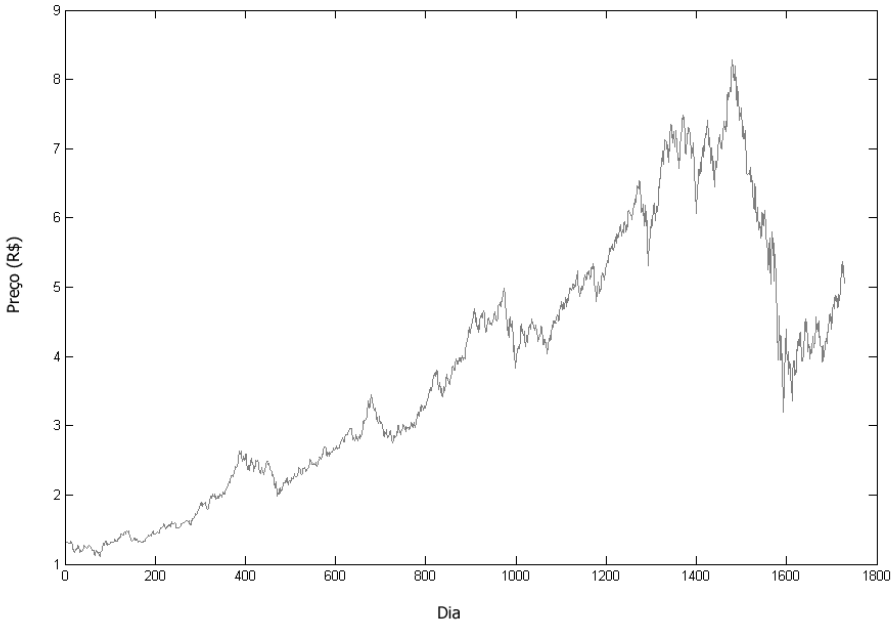

Figura 14: Série de preços das cotas - Fundo de ações IBrX

4.2 Ferramentas Utilizadas

Para a realização de toda uma extensa série de experimentos, e tornar real o modelo teórico desenvolvido, vários programas foram criados e outros adaptados à necessidade da pesquisa a partir de código existente, desenvolvido anteriormente por pesquisadores na área de redes neurais. Todo o código foi implementado na linguagem C/C++, baseado em algoritmos clássicos e difundidos.

Para o caso específico das máquinas de vetor de suporte, foi utilizada a biblioteca LIBSVM (CHANG; LIN, 2001), desenvolvida por pesquisadores da *National Taiwan University* e escolhida entre diversas outras implementações de outros pesquisadores ao redor do mundo (CANU et al., 2005) (HOCHREITER; OBERMAYER, 2006), devido aos resultados satisfatórios comprovados em diversos artigos publicados, além da facilidade na utilização e personalização de seu código aberto de acordo com a necessidade e os rumos da pesquisa.

A LIBSVM é uma biblioteca integrada para classificação (C-SVC, *nu*-SVC), regressão (ε-SVR, *nu*-SVR) e estimação de distribuição (*one-class* SVM), baseadas em máquinas de vetor de suporte. Possui interfaces desenvolvidas em C/C++, Java, MATLAB, Perl e várias outras linguagens. Sua utilização é simples e conta com a possibilidade de ajuste para os parâmetros livres (C, ε), tipos de função de kernel, etc. No decorrer deste trabalho somente foi utilizada a versão escrita em C++ da biblioteca.

Além disso, a versão R2008a da conhecida ferramenta MATLAB da empresa *Mathworks* foi amplamente utilizada, tanto para fins de testes iniciais de eficiência com o perceptron de multiplas camadas (MLP) e mapas auto-organizáveis (SOM) quanto para a demonstração gráfica de resultados e cálculos diversos, incluindo medidas de erro e análise estatística.

Para os testes finais, foram utilizadas as versões da MLP, SOM e SVM implementadas em C/C++. Todos os gráficos com os resultados dos experimentos para o modelo final foram confeccionados através da ferramenta livre *gnuplot*.

4.3 Testes Preliminares

A fase inicial de experimentos, no desenvolvimento do trabalho, foi concentrada em testes com as técnicas mais simples de previsão, objetivando a familiarização com as ferramentas que seriam utilizadas na implementação e validação do modelo definitivo e com os modelos básicos de redes neurais, ainda sem a utilização de arquiteturas hierárquicas e com poucas modificações no código-fonte.

Os testes seguiram um curso progressivo, iniciando de maneira mais ingênua e incorporando gradativamente características que viriam a ser importantes no modelo definitivo, como a realimentação da rede e a utilização de uma janela com os dados de entrada. Além disso, os experimentos demonstraram que testes para previsão com a série "pura", sem processamento prévio, utilizando apenas os valores de preço das cotas, não são promissores e que um estudo cuidadoso da série é necessário para a extração e consequente exploração de suas características mais importantes.

4.3.1 Prevendo um Passo a Frente

Inicialmente, foram realizados testes simples para previsão de um dia a frente, tendo como entrada da rede apenas um mínimo de informação - o dia atual - e utilizando a

máquina de vetor de suporte básica da biblioteca LIBSVM. Neste primeiro teste a série utilizada foi a de preços das cotas do fundo de investimento, sem processamento prévio e nenhum tipo de normalização.

A Figura 15 nos mostra que os resultados foram razoáveis, dado o horizonte curto adotado para a previsão. Nela, a curva em azul representa os dados reais e a curva em verde nos mostra os valores previstos. O teste cumpriu sua finalidade, que era demonstrar a funcionalidade da ferramenta de previsão LIBSVM.

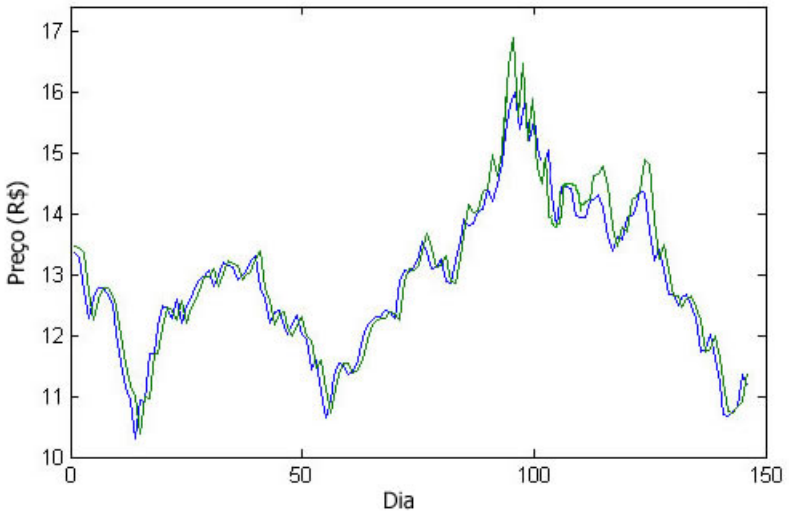

Figura 15: Previsão com SVM - Um passo a frente

A seguir, algumas modificações começaram a ser realizadas no código com o objetivo de facilitar a previsão, adicionando mais informação referente ao contexto dos dados, relativa a seu passado e estado atual. Na primeira delas, acrescentou-se um termo *alpha* (α), conhecido como integrador de tempo, através do qual a rede seria capaz de manter informações de seu histórico recente. Com a inserção do novo termo, a intenção é adicionar informação relevante para a máquina: a cada passo, a saída $y(t-1)$ obtida no passo anterior é multiplicada por α e somada à nova entrada $x(t)$, de maneira a contribuir para a tomada de decisão através da ampliação da memória da rede.

Finalmente, uma nova entrada $x'(t)$ é gerada de acordo com a equação (4.1) abaixo:

$$x'(t) = x(t) + \alpha * y(t-1) \tag{4.1}$$

O resultado, mesmo que contra as expectativas, foi uma diminuição na eficiência e aumento no erro de previsão da rede, como ilustrado claramente na Figura 16, abaixo:

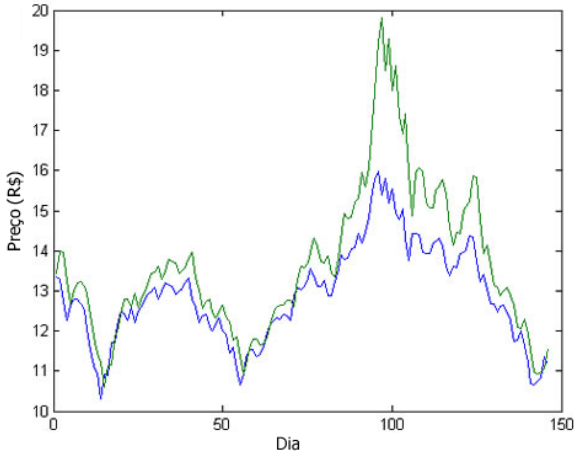

Figura 16: Previsão com SVM - Um passo a frente com histórico (α)

4.3.2 Aumentando o Horizonte de Previsão

Seguindo o curso natural dos testes e antecipando o objetivo real da pesquisa, o passo seguinte foi o aumento do horizonte de previsão. A implementação deste aumento se deu através do conceito de realimentação da rede, alcançada através da utilização sucessiva do valor previsto como entrada para a previsão do próximo. Mais uma modificação foi realizada no código, de modo que o valor previsto atuasse como entrada da rede no próximo passo, como segue:

$$x(t) = y(t-1) \tag{4.2}$$

O erro de previsão se propaga e se potencializa rapidamente, resultando em uma perda completa de contexto pela rede, como vemos na Figura 17, abaixo:

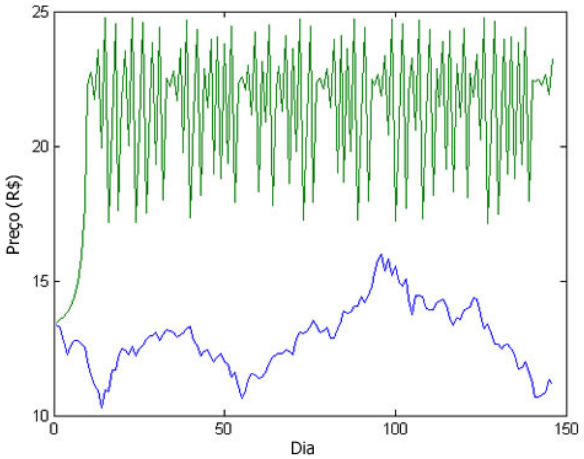

Figura 17: Previsão com SVM - Rede realimentada

4.3.3 Janela Contínua de Entrada

Uma alternativa à inclusão de histórico através do integrador de tempo, mas ainda com a finalidade de aumentar a memória do modelo, é a utilização de uma janela contínua de entrada. Para um determinado dia, são utilizados como entrada os valores de preço das cotas nos n dias antecedentes, resultando em padrões de entrada com n componentes.

Janelas com diferentes quantidades (n) de componentes foram testadas. No entanto, utilizada em conjunto com a realimentação e dando prosseguimento à tentativa de previsão com um horizonte mais extenso, esta técnica também não obteve melhorias significativas. A Figura 18 mostra o resultado da previsão utilizando como entrada uma janela de oito dias.

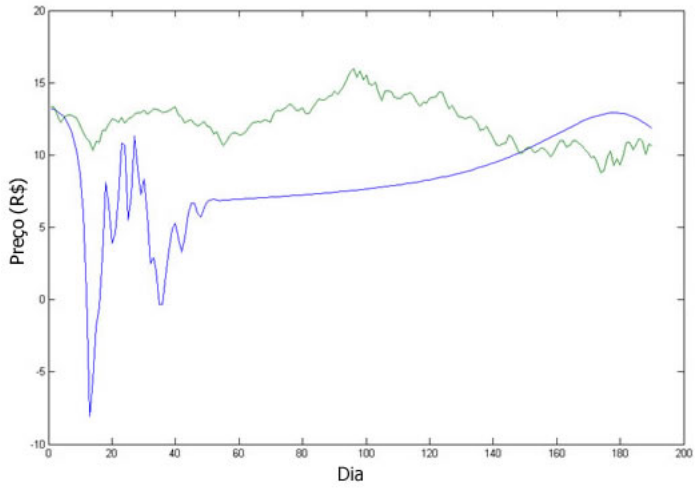

Figura 18: Previsão com SVM - Rede realimentada e Janela com 8 dias

4.3.4 Testes com MLP

Com a finalidade de verificar os resultados obtidos pela SVM, foram também efetuados testes com um modelo já consagrado na literatura: os perceptrons de múltiplas camadas, com algoritmo de retro-propagação (*back-propagation*), mais conhecidos como MLP (do inglês *Multi-Layer Perceptron*). Como podemos ver nas Figuras 19a e 19b, os resultados foram semelhantes à SVM para a previsão de um passo a frente e para a realimentação. Este fato nos leva a considerar uma análise mais apurada e profunda da série utilizada, para a obtenção de resultados mais significativos.

Outros testes foram realizados utilizando o modelo MLP, incluindo tentativas de janelamento e integrador de tempo (*alpha* e histórico), mas os resultados foram igualmente insatisfatórios.

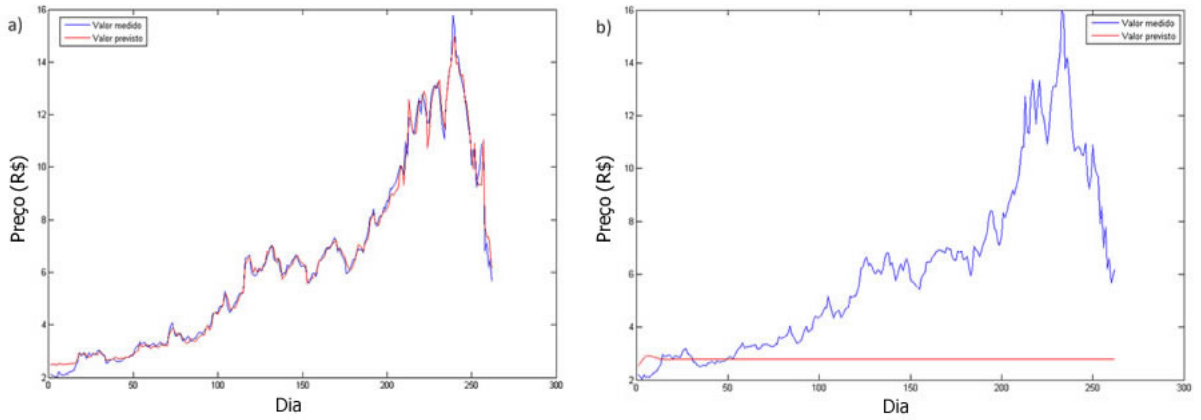

Figura 19: Previsão com MLP - a) Um passo a frente; b) Realimentação

4.4 Estudo da Série Temporal

4.4.1 Série de Retorno

A grande maioria dos estudos financeiros envolve taxa de retorno ao invés de preço. Como discutido no capítulo 2, é comum supor que a série de retorno de um ativo é fracamente estacionária (TSAY, 2002), e obtida através de uma transformação ou normalização logarítmica nos dados da série de preços, que é não-estacionária e portadora de forte componente de tendência.

Supondo o retorno de um investimento como uma coleção de variáveis no tempo, obtém-se uma nova série temporal, e a análise linear de séries temporais sempre nos proporciona um conjunto de ferramentas para o estudo de sua dinâmica através de propriedades que incluem estacionariedade, dependência dinâmica, função de autocorrelação, etc. Podemos dizer que, além de representar mais claramente a oportunidade de investimento, as séries de retorno exibem também características estatísticas mais atrativas que séries de preços, facilitando a análise e tornando o aproveitamento dessas caracterísicas mais direto no processo de previsão.

No caso de nossa série, utilizou-se o retorno composto contínuo (ou *log return*), caracterizado pelo logaritmo natural do retorno simples bruto, e calculado pela fórmula (4.3) abaixo:

$$R_t = ln\left(\frac{P_t}{P_{t-1}}\right) \tag{4.3}$$

Através do cálculo da taxa de retorno para todos os dias disponíveis na série preços,

obtém-se uma nova série temporal de características completamente distintas. Como podemos observar na Figura 20, a aplicação da transformação resulta em uma série (fracamente) estacionária de aspecto horizontal, com média próxima a zero e comportamento cíclico, caracterizado pelo aparecimento de *clusters* de volatilidade mais alta ou mais baixa, que seguem a tendência de aquecimento ou calmaria no mercado financeiro. Esta é a série utilizada no decorrer dos testes finais para o modelo hierárquico desenvolvido.

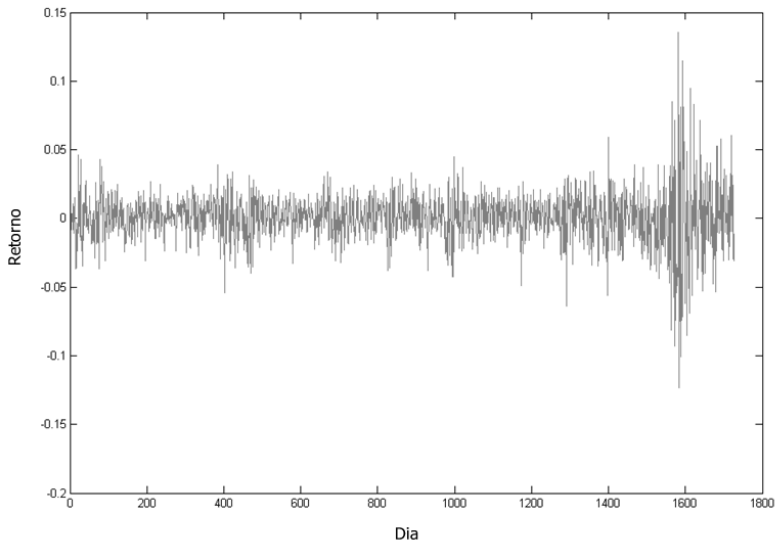

Figura 20: Série de retorno contínuo - Fundo de investimento IBrX

4.4.2 Função de Autocorrelação

Em uma série fracamente estacionária, se a dependência linear entre o valor atual e os valores passados for de interesse, pode-se utilizar o conceito de autocorrelação (TSAY, 2002). Correlações entre a variável de interesse e seus valores passados - ou autocorrelação - são um dos focos da análise de séries temporais. Seu objetivo é quantificar as relações de dependência linear entre o valor de interesse e seu passado próximo, de maneira a explorar esta informação no processo de previsão. O fato de um retorno diário r_t possuir uma correlação significativa com a taxa de retorno no dia anterior, por exemplo, indica que este ultimo valor (chamado de *lag-1*) pode ser útil na previsão de r_t.

A função de autocorrelação é descrita pela equação (4.4), onde $\hat{\rho}_l$ é o coeficiente de correlação para *lag-l*, r_t é o valor da taxa de retorno no tempo t e $\bar{r}$ é o valor médio da série. Sua utilização na série de retorno contínuo do fundo de investimentos, para os vinte dias anteriores, resulta no gráfico mostrado na Figura 21.

$$\hat{\rho}_l = \frac{\sum_{t=l+1}^{T}(r_t - \bar{r})(r_{t-l} - \bar{r})}{\sum_{t=1}^{T}(r_t - \bar{r})^2}, \quad 0 \leq l < T - 1 \tag{4.4}$$

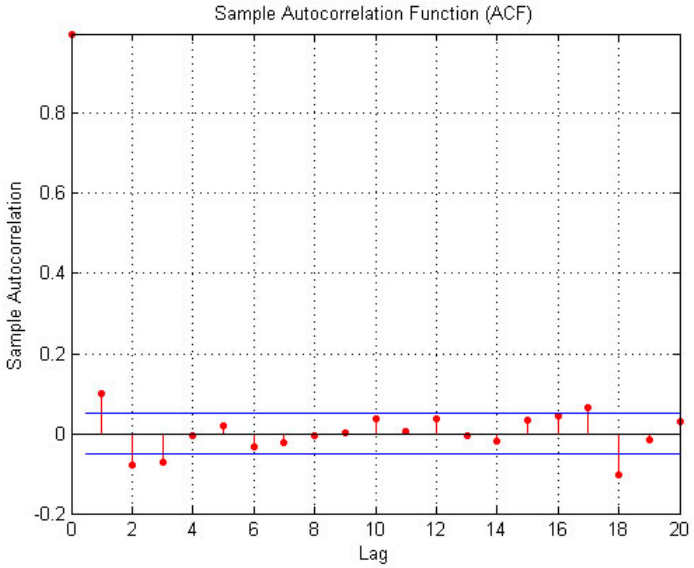

Figura 21: Autocorrelação - Série de retorno

O sinal do coeficiente de correlação obtido (+ ou -) indica a direção do relacionamento entre os elementos da série. Em termos práticos, dizemos que em caso de sinal positivo, os elementos tendem a aumentar ou diminuir juntos e, no caso contrário, um deles aumenta enquanto o outro diminui. Se o valor é próximo a zero, cada um deles segue caminhos independentes, indicando que a magnitude do coeficiente de correlação é uma medida de força na dependência entre elas.

Através da análise do correlograma, percebe-se que alguns valores do passado imediato da série possuem correlações mais significativas ($\geq 5\%$) com o valor atual do que outras: *lags* 1, 2, 3, 17 e 18. Esta correlação, calculada a partir de todos os dados da série, é válida para todo o período que ela contempla.

Os valores obtidos pela aplicação da fórmula transformam-se imediatamente em fortes candidatos para a construção dos padrões de entrada do modelo, pois apresentam apenas informação relevante sobre o passado da série. Sua utilização possibilita a construção de um contexto menos ruidoso pelo modelo neural, funcionando como um filtro que, retendo dados irrelevantes, torna viável a previsão do comportamento da série com uma margem de erro consideravelmente pequena.

4.5 Modelo Hierárquico

4.5.1 Características Fundamentais

Muitos testes foram executados com arquiteturas individuais de redes neurais dos tipos SVM e MLP, ambas com processos de aprendizagem supervisionados. Na nova etapa de experimentos passou-se a utilizar o modelo neural hierárquico (HNM, de *hierarchical neural model*), composto basicamente por um estágio não-supervisionado, representado por um mapa auto-organizável (SOM), conectado a um estágio supervisionado, responsável pela saída definitiva do modelo e representado por uma máquina de vetor de suporte (SVM). A Figura 22 representa de maneira simplificada o modelo proposto, cujas características estão descritas detalhadamente no capítulo 3.

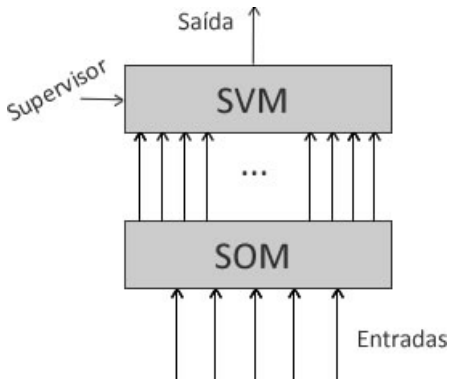

Figura 22: Modelo Hierárquico: SOM e SVM

Como dito anteriormente, os modelos hierárquicos são capazes de armazenar com grande eficiência a memória de eventos passados devido à segmentação da série temporal em diferentes contextos, traduzindo um padrão de grande complexidade para um conjunto de características primitivas através das quais também é possível identificar o padrão como um todo.

Portanto, o objetivo do modelo desenvolvido é extrair e armazenar em sua estrutura o máximo de informações do grupo de dados fornecidos como entrada, formados por dados passados da série, criando um mapa de características na saída da SOM. Finalmente, ao apresentar à entrada da máquina de vetor de suporte este novo conjunto de dados, com propriedades mais evidentes da série de retorno, espera-se obter um desempenho superior de previsão em relação aos outros modelos mais simples.

4.5.2 Escolha do Modelo Hierárquico Ideal

Devido aos resultados obtidos nos testes preliminares, foram abandonadas as tentativas de previsão da série de preços, concentrando os esforços na série de retorno do fundo de investimento IBrX e na escolha do melhor modelo para a previsão deste tipo de dados. Esta fase se deu com a realização de testes utilizando as arquiteturas já apresentadas como a máquina de vetor de suporte (SVM) pura e do modelo hierárquico SOM + SVM (HNM), além de um novo tipo, caracterizado pela adição de outra rede SOM ao modelo hierárquico resultando em uma arquitetura SOM + SOM + SVM, baseada em um modelo já existente e bem-sucedido (CARPINTEIRO; BARROW, 1996), utilizado para problemas de classificação. A análise dos resultados obtidos por cada uma das arquiteturas será utilizada na escolha do modelo definitivo, utilizado nos testes finais.

Nesta fase dos experimentos, passou-se também a adotar como entrada os valores do passado imediato da série que segundo análise dos cálculos e gráfico de autocorrelação (Figura 21) apresentaram maior nível de correlação com o valor atual, resultando em padrões de entrada formados pelo valor do retorno no dia imediatamente anterior, 2, 3, 17 e 18 dias atrás do dia atual (*lags* 1, 2, 3, 17 e 18). Os dados não sofreram nova normalização devido ao fato de já se encontrarem em uma faixa bem definida de valores, entre aproximadamente -0,15 e 0,15. A Figura 23 ilustra o processo de inserção dos parâmetros de entrada:

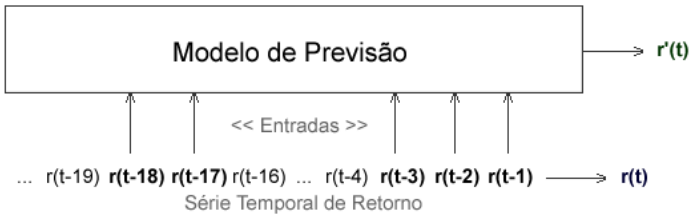

Figura 23: Entradas do modelo de previsão

A escolha destes valores de entrada possibilita a utilização apenas da informação que é realmente relevante para a rede, aumentando a eficiência da previsão, descartando dados pouco importantes ou mesmo prejudiciais e diminuindo o tempo de processamento. Todos os testes apresentados a seguir foram realizados para um horizonte de previsão de vinte dias com uma rede realimentada, ou seja, o valor previsto é apresentado como uma das entradas (*lag*-1) para a próxima previsão.

A medida de erro utilizada para a comparação entre os modelos é o "erro absoluto médio" ou MAE (do inglês *mean absolute error*), obtida através da fórmula descrita pela

equação (4.5), onde y_i é o valor real, $\hat{y}_i$ representa o valor previsto e n é a quantidade de previsões realizadas.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.5}$$

A definição do MAE para a análise dos modelos desta fase não compromete o estudo, uma vez que o objetivo é apenas quantificar e comparar seus desempenhos. Os testes realizados para o modelo escolhido e sua comparação com outras arquiteturas mais tradicionais serão tratados adiante, utilizando uma medida de erro percentual (MAPE), que torna os resultados mais claros e interpretáveis.

4.5.2.1 SVM Pura

Foram realizados vários testes para a SVM pura e os resultados foram similares na maioria deles. No melhor resultado obtido, ilustrado pela Figura 24, os parâmetros C e ε da ferramenta LIBSVM foram regulados para os valores 1000 e 0,001, respectivamente. Os resultados foram, em geral, muito ruins e a previsão não seguiu a tendência da série, apresentando um aspecto praticamente plano e um erro médio absoluto mínimo de **0,0181**. O gráfico mostra os valores reais de retorno em azul e os valores previstos em vermelho.

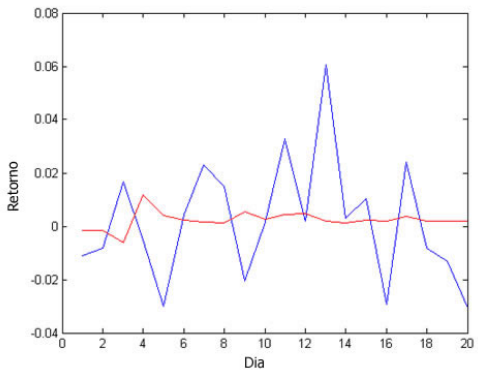

Figura 24: SVM Pura: Melhor resultado

4.5.2.2 HNM (SOM + SVM)

Nesta fase de testes, os dados de entrada acionam um mapa auto-organizável gerando m valores de saída, sendo m o número de nós computacionais ou neurônios do mapa. As m saídas do mapa servem como entrada para uma máquina de vetor de suporte, que realiza a função do treinamento supervisionado. Foram testadas várias configurações com

diferentes dimensões de mapas e vários valores para os parâmetros C e ε da SVM. Entre elas destacam-se algumas:

1. Rede SOM de 20x20 neurônios, com raio da gaussiana igual a 0,007 e SVM com $C = 1000$ e $\varepsilon = 0{,}001$. Foi obtido um erro médio de **0,0154** e o resultado segue a tendência do gráfico, com uma falha sensível apenas no 13º dia (Figura 25).

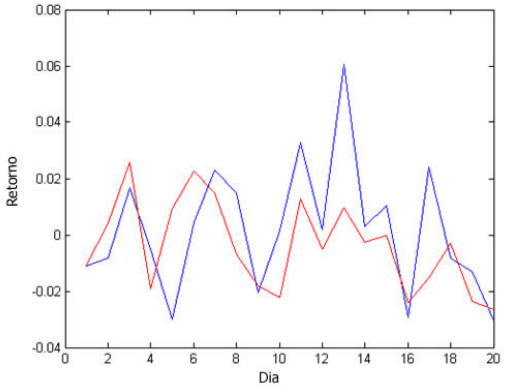

Figura 25: SOM + SVM: Primeira previsão

2. Rede SOM de 20x20 neurônios, com raio da gaussiana igual a 0,008 e SVM com $C = 1000$ e $\varepsilon = 0{,}001$. Foi obtido um erro médio de **0,0204**, apontando que o aumento no raio da gaussiana interferiu negativamente no resultado (Figura 26).

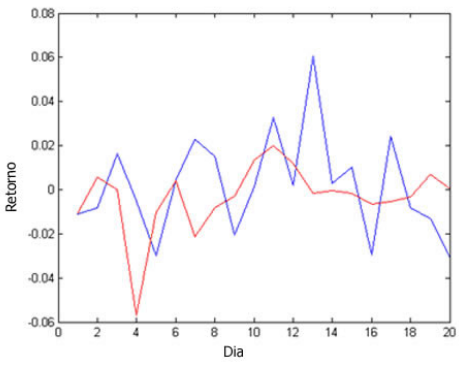

Figura 26: SOM + SVM: Segunda previsão

3. Rede SOM de 18x18 neurônios, com raio da gaussiana igual a 0,007 e SVM com $C = 1000$ e $\varepsilon = 0{,}001$. Erro médio de **0,0143** - O menor erro obtido pelos modelos testados nesta fase (Figura 27).

4. Rede SOM de 18x18 neurônios, com raio da gaussiana igual a 0,007 e SVM com $C = 500$ e $\varepsilon = 0{,}001$. Erro médio de **0,0157** - Desempenho semelhante ao anterior, com erro levemente mais alto (Figura 28).

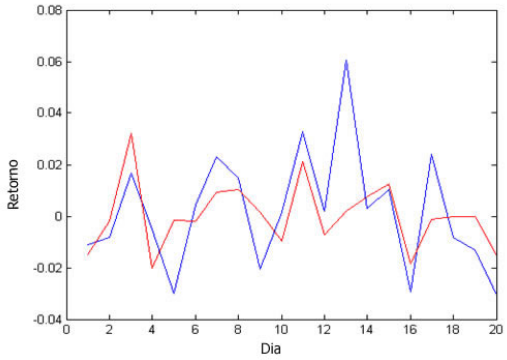

Figura 27: SOM + SVM: Terceira previsão

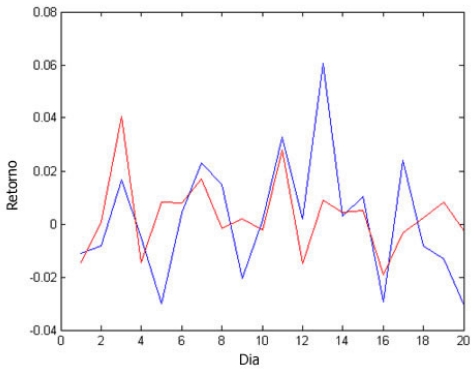

Figura 28: SOM + SVM: Quarta previsão

4.5.2.3 SOM + SOM + SVM

Por fim, mais uma rede SOM foi incorporada ao modelo. A utilização de duas redes do tipo SOM - uma sobre a outra - não é uma idéia original, tendo sido utilzada com sucesso para problemas de classificação (CARPINTEIRO; BARROW, 1996), mas não para regressão. Aqui, o intuito é capturar o máximo das características da entrada e fornecer um grupo de dados cada vez mais significativo para a máquina de vetor de suporte. Foram testadas novamente várias combinações de parâmetros e dimensões das redes SOM. Destacam-se alguns resultados:

1. Rede SOM de 15x15 e raio da gaussiana 0,003, seguida de outra rede SOM de 18x18 com raio da gaussiana 30,0 e SVM com $C = 2000$ e $\varepsilon = 0{,}001$. Erro médio obtido de **0,0147** (Figura 29).

2. Rede SOM de 20x20 e raio da gaussiana 0,003, seguida de outra rede SOM de 25x25 com raio da gaussiana 30,0 e SVM com $C = 2000$ e $\varepsilon = 0{,}001$. Erro médio obtido de **0,0175** - Maior que o teste anterior, mostrando que o simples fato do aumento

das redes SOM não necessariamente afeta de forma positiva a previsão (Figura 30).

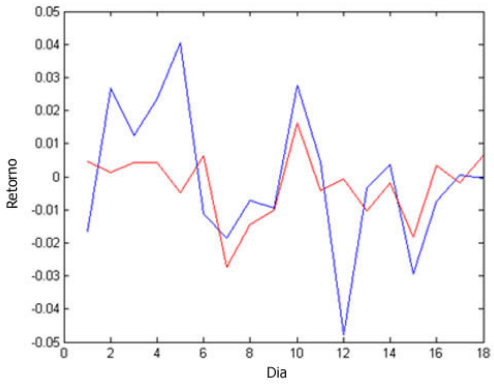

Figura 29: SOM + SOM + SVM: Primeira previsão

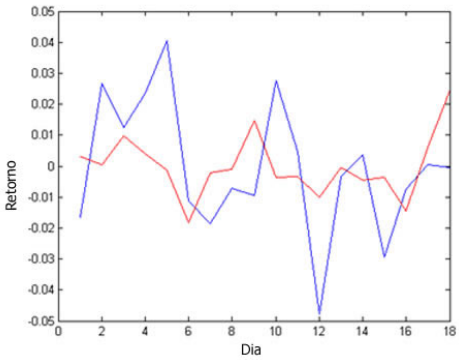

Figura 30: SOM + SOM + SVM: Segunda previsão

4.5.3 Modelo Escolhido

Através dos resultados, é possível perceber que o modelo neural hierárquico HNM, formado por um mapa auto-organizável e uma máquina de vetor de suporte, apresentou os melhores resultados, indicando a linha de trabalho com maior potencial e o modelo de rede neural a ser explorado nos passos seguintes. O modelo que se utiliza de duas redes SOM abaixo da SVM também apresentou resultados interessantes mas não se destacou em relação ao modelo anterior, mostrando um aumento na complexidade - e tempo de execução - que não se traduz em um benefício de desempenho da mesma magnitude nos resultados da previsão.

4.6 Comparação com Modelos Estabelecidos

Para a validação do modelo hierárquico desenvolvido e observação de seu comportamento diante de modelos neurais tradicionais, realizou-se uma extensa bateria de testes. Os modelos utilizados como parâmetros na comparação de desempenho foram uma rede neural do tipo perceptron de múltiplas camadas (MLP) e uma máquina de vetor de suporte (SVM).

Esta fase consiste na execução de dez testes para cada um dos modelos, sendo que cinco deles se encontram situados em períodos de estabilidade do mercado e outros cinco em períodos turbulentos e mais instáveis (diferentes volatilidades). Em todos eles o conjunto de treinamento utilizado possui 1848 padrões de entrada em forma de janela com *lags* 1, 2, 3, 17 e 18, referentes à série de retorno do fundo de investimento IBrX do Banco do Brasil no período entre 30 de julho de 2002 e 31 de dezembro de 2009.

As datas dos períodos previstos estão detalhadas na Tabela 1, e é importante ressaltar que os vinte pontos a serem previstos em cada experimento foram obviamente retirados de seus respectivos conjuntos de treinamento, visando conhecer a capacidade de generalização do modelo proposto.

Tabela 1: Períodos Previstos - B: Baixa volatilidade; A: Alta volatilidade

Experimento	**Início**	**Final**
1B	24 Junho 2003	21 Julho 2003
2B	22 Novembro 2004	17 Dezembro 2004
3B	23 Agosto 2005	20 Setembro 2005
4B	28 Maio 2007	25 Junho 2007
5B	11 Setembro 2009	08 Outubro 2009
1A	28 Janeiro 2005	28 Fevereiro 2005
2A	19 Maio 2006	16 Junho 2006
3A	18 Janeiro 2008	18 Fevereiro 2008
4A	03 Setembro 2009	30 Setembro 2008
5A	14 Outubro 2009	11 Novembro 2009

A medida utilizada para a definição de períodos de estabilidade (baixa volatilidade) e instabilidade (alta volatilidade) possui relação com o cálculo do desvio padrão da série, que se apresenta como uma boa medida de volatilidade para a série (YANG; CHAN; KING, 2002). O aparecimento de variações mais bruscas e frequentes no passado próximo de um dia indicam fortemente que este dia se inserirá num período de instabilidade, da mesma maneira que um passado calmo leva a um futuro imediato também estável. Segundo Tsay

(2002), este raciocínio é válido, pois a volatilidade evolui continuamente no tempo, e saltos de volatilidade são raríssimos.

Por este motivo, criou-se um índice binário para a determinação quantitativa da volatilidade num determinado instante da série, ou seja, a medida de estabilidade num dado dia. O índice é definido através da comparação entre o desvio padrão da série como um todo (total) e o cálculo deste mesmo desvio padrão utilizando apenas os vinte valores imediatamente anteriores ao dia analisado (instantâneo), ambos calculados de acordo com a fórmula da Equação (4.6), onde n é a quantidade de amostras e $\bar{x}$ é o cálculo da média da série. Quando o valor do "desvio padrão instantâneo" é superior ao "desvio padrão total" o dia é marcado como instável (1) e do contrário, como estável (-1). Assim, foram determinados períodos de alta e baixa volatilidade.

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4.6}$$

A informação sobre volatilidade é privilegiada e constitui conhecimento adicional sobre o comportamento da série no momento da previsão. Por este motivo, o valor do índice de volatilidade foi incorporado à entrada da rede, formando padrões compostos por 6 valores: 5 entradas de *lag* com valores passados da taxa de retorno do fundo de investimento e um valor relativo à volatilidade instantânea (1 ou -1).

Para a medida de erro nesta fase, foi abandonado o MAE (erro absoluto médio), utilizado nos testes preliminares, e adotado o "erro absoluto médio percentual" ou MAPE (do inglês *mean absolute percentual error*), calculado de acordo com a Equação (4.7). Esta mudança se deve ao fato de que o MAPE apresenta uma medida mais clara e intuitiva, facilitando a interpretação dos resultados e comparação entre métodos. Por exemplo, dizer que um experimento resultou em um MAPE de 5% é muito mais significativo do que informar que o erro total foi de 0,0143. O MAPE se apresenta como uma medida livre de escala e ideal para o objetivo desta fase do trabalho.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4.7}$$

Todos os parâmetros disponíveis para ajuste foram modificados e testados com o objetivo de obter os melhores resultados possíveis para cada um dos modelos, e não somente no modelo hierárquico. As tabelas 2, 3 e 4 a seguir exibem os valores utilizados como parâmetros durante os experimentos cujos resultados foram mais eficientes.

Tabela 2: Parâmetros: Modelo Hierárquico HNM

SOM	
Dimensões	50 x 50
Entrada	6 entradas
Raio da Vizinhança	5
Treinamento (*Coarse-Mapping*)	
Épocas	400
Taxa de Aprendizagem Inicial	0,5
Raio da Vizinhança Inicial	50
Treinamento (*Fine-Tuning*)	
Épocas	1000
Taxa de Aprendizagem Inicial	0,01
Raio da Vizinhança Inicial	1
SVM	
Tipo	ε-SVR
Função de *Kernel*	Função de base radial (RBF)
C	1000
ε	0,005

Tabela 3: Parâmetros: MLP

MLP	
Unidades Ocultas	30
Taxa de Aprendizagem Inicial	0,1
Momento	0,7
Épocas de Treinamento	850

Tabela 4: Parâmetros: SVM Pura

SVM	
Tipo	ε-SVR
Função de *Kernel*	Função de base radial (RBF)
C	1000
ε	0,007

A Tabela 5 apresenta os melhores resultados alcançados por cada um dos modelos utilizados nos testes de previsão, em relação ao erro médio total mostrado em sua última linha. Os valores apresentados na tabela estão expressos em termos de erro percentual absoluto médio, ou MAPE, e os índices A e B diferenciam os testes realizados em períodos de alta e baixa volatilidade, respectivamente.

Tabela 5: Resultados: Comparação entre modelos; Valores de MAPE

Experimento	MLP	SVM	HNM
1B	1,5541	**1,1785**	4,1581
2B	2,9294	2,4037	**0,9902**
3B	4,3077	**2,6339**	5,8239
4B	1,7597	**1,1738**	1,9167
5B	4,1239	**2,0040**	2,5711
1A	6,1807	**5,8912**	7,1402
2A	9,1115	8,1148	**5,0196**
3A	3,5636	2,8835	**2,1515**
4A	10,6918	10,5901	**6,2069**
5A	2,3696	2,8112	**1,9342**
Erro médio	**4,6592**	**3,9685**	**3,79124**
Desvio Padrão	3,0988	3,1786	2,1554

As figuras a seguir mostram graficamente os resultados obtidos para os testes. Em cada uma delas estão ilustrados os resultados da previsão de cada um dos três modelos (HNM em vermelho, MLP em verde e SVM em azul) mais a série original, representada por uma linha pontilhada.

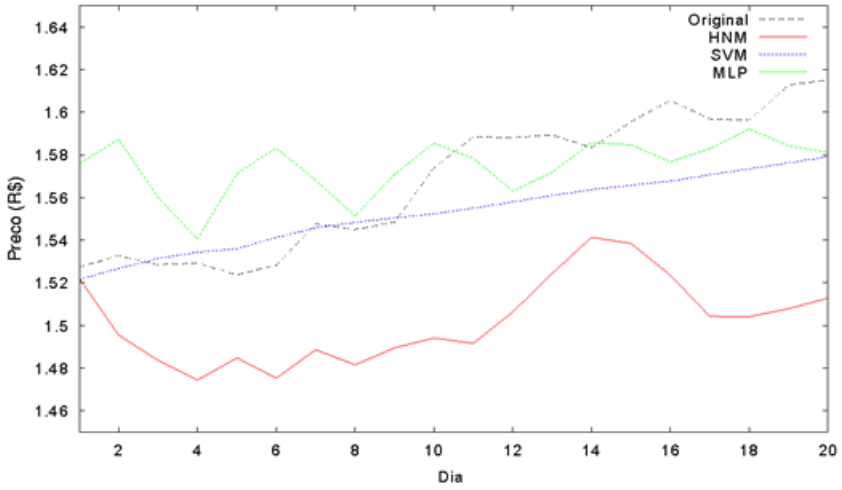

Figura 31: Previsão: Período 1, Baixa volatilidade (1B)

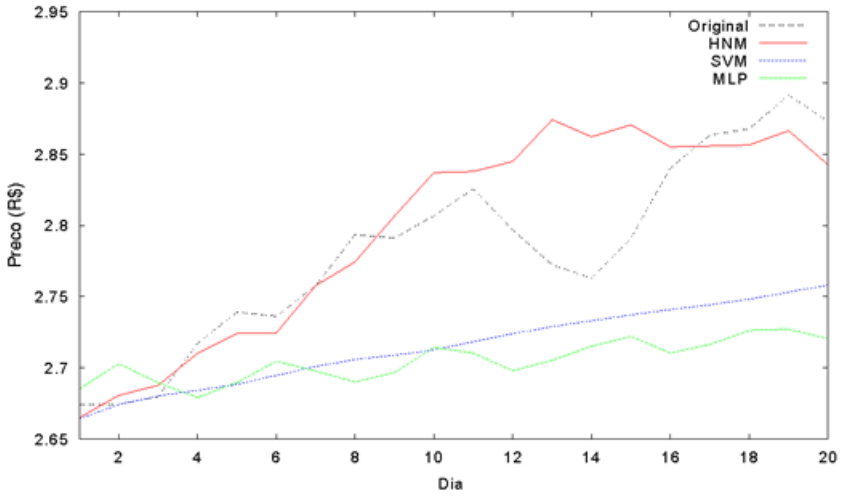

Figura 32: Previsão: Período 2, Baixa volatilidade (2B)

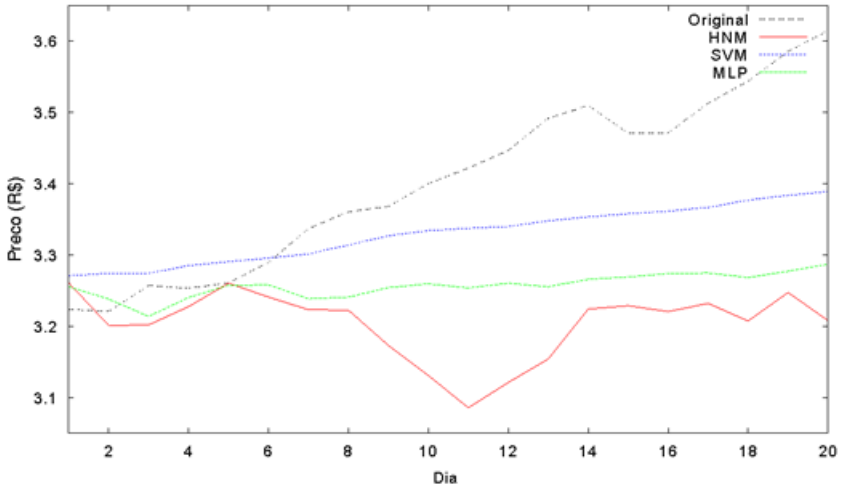

Figura 33: Previsão: Período 3, Baixa volatilidade (3B)

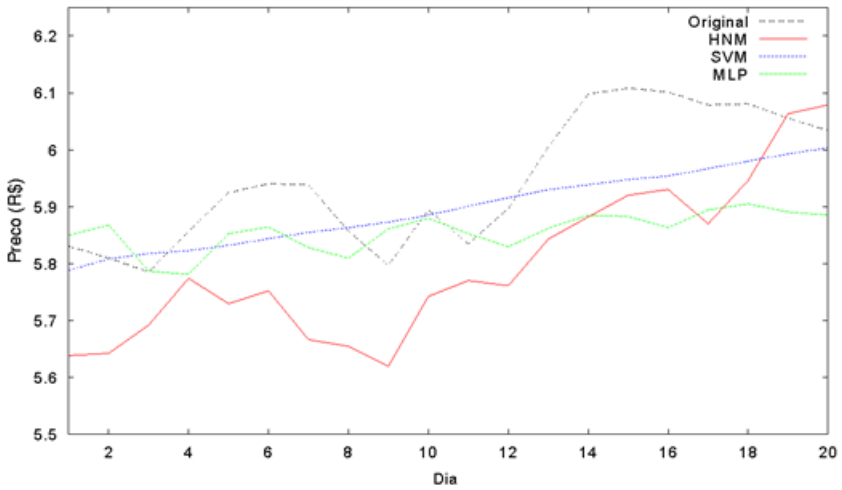

Figura 34: Previsão: Período 4, Baixa volatilidade (4B)

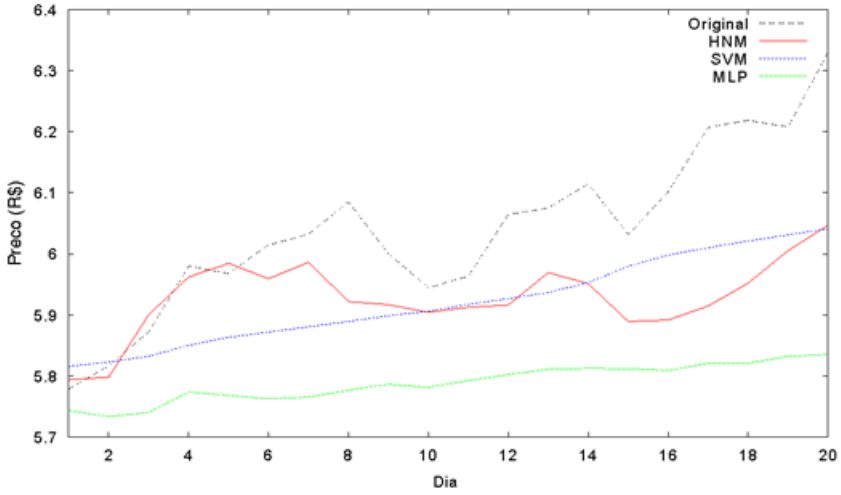

Figura 35: Previsão: Período 5, Baixa volatilidade (5B)

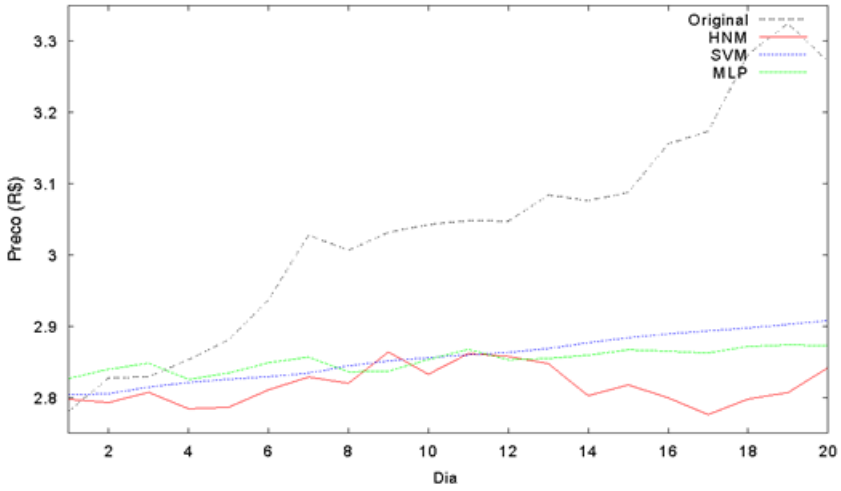

Figura 36: Previsão: Período 1, Alta volatilidade (1A)

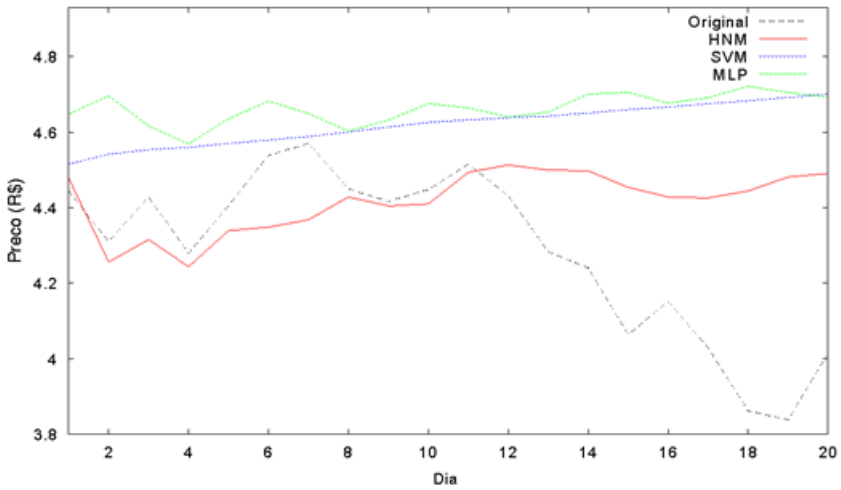

Figura 37: Previsão: Período 2, Alta volatilidade (2A)

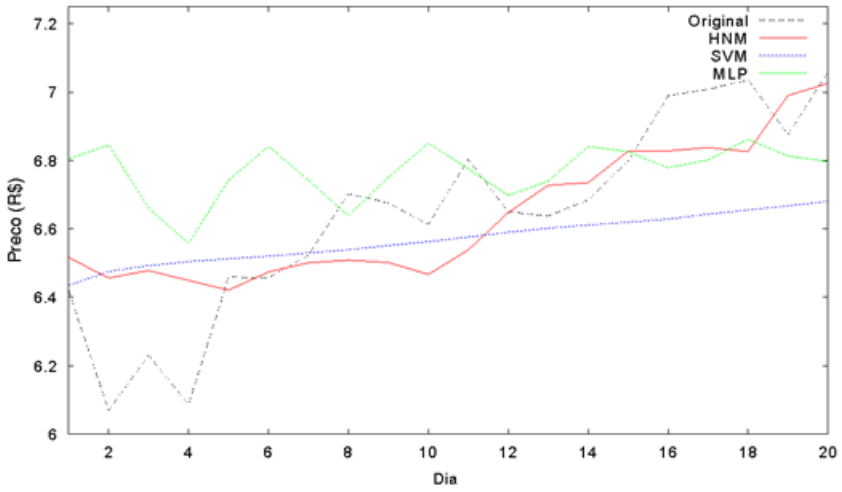

Figura 38: Previsão: Período 3, Alta volatilidade (3A)

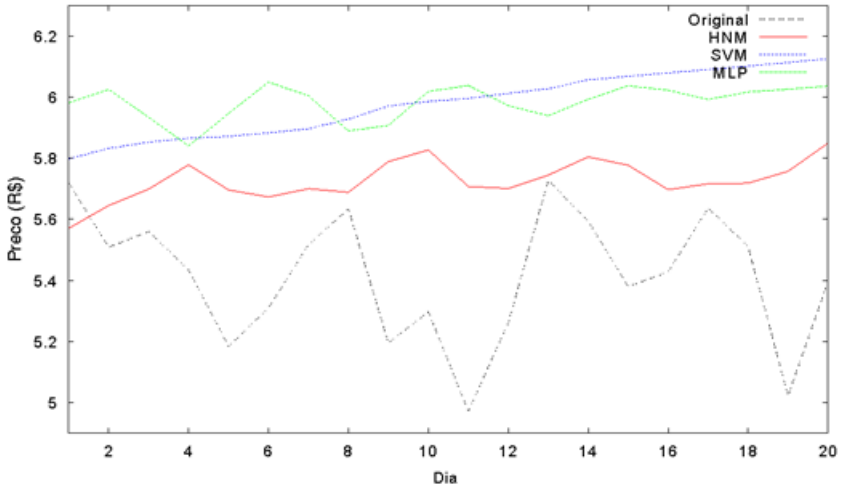

Figura 39: Previsão: Período 4, Alta volatilidade (4A)

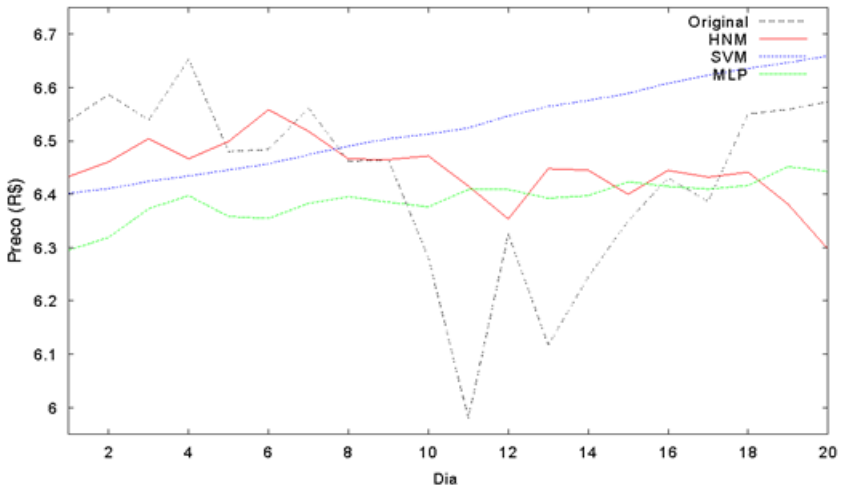

Figura 40: Previsão: Período 5, Alta volatilidade (5A)

Através dos resultados obtidos individualmente para cada um dos exemplos, mais a análise visual dos gráficos plotados com dados diários, é possível tirar algumas conclusões sobre o resultado geral obtido.

O erro médio, ou seja, a média aritmética dos erros obtidos em cada um dos dez experimentos, foi menor nos testes realizados com o modelo hierárquico do que nos outros modelos. Percentualmente, houve uma melhora de aproximadamente 4,5% em relação à máquina de vetor de suporte pura e 18,5% em relação à rede do tipo MLP. Além disso, o modelo hierárquico apresentou resultados superiores aos dos demais modelos em cinco dos testes - exatamente a metade (2B, 2A, 3A, 4A e 5A) - e obteve resultados significativamente inferiores apenas em dois deles (testes 1 e 3 de baixa volatilidade, nas figuras 31 e 33). É possível notar ainda que os resultados obtidos para testes em períodos de alta volatilidade, ou seja, grande instabilidade, foram bastante superiores para o modelo hierárquico, na maioria dos casos.

Outro fator de grande importância pode ser percebido através da análise gráfica dos dados obtidos. O modelo hierárquico acompanha a tendência da variação de preços e aprende a dinâmica de uma série financeira com maior fidelidade que os demais modelos, cujas previsões se traduzem em gráficos com aspecto plano e deficiente de maior sensibilidade a pequenas oscilações do mercado. Esta característica acabou por se tornar uma vantagem para os modelos mais simples na previsão em períodos de baixa volatilidade, onde as oscilações são brandas, mas comprometeu de maneira expressiva o desempenho destes modelos em períodos de maior volatilidade.

Finalmente, devido à observação das características exibidas pela série temporal financeira, e buscando um resultado que refletisse melhor a capacidade de previsão do modelo hierárquico, novos experimentos foram realizados com o objetivo de explorar mais adequadamente as variações de volatilidade no decorrer do tempo e as claras diferenças existentes entre estes pequenos ciclos de maior ou menor turbulência.

4.7 Explorando a Volatilidade

4.7.1 Fundamentação Teórica

No decorrer dos estudos, a importância de uma das características da série do fundo de investimentos em especial tornou-se bastante clara: sua mudança de comportamento de tempos em tempos, tornando-se mais ou menos volátil, ou seja, variando com maior

ou menor intensidade.

Comprovada através do cálculo de desvio padrão mostrado na seção anterior, esta característica também pode ser identificada facilmente no gráfico do valor diário da taxa de retorno no tempo, mostrado na Figura 41. Nele, o sinal em azul representa a taxa de retorno diária, a linha horizontal em verde é o devio padrão da série como um todo e, em vemelho, está representado o desvio padrão instantâneo, calculado apenas com os valores de retorno dos últimos vinte dias.

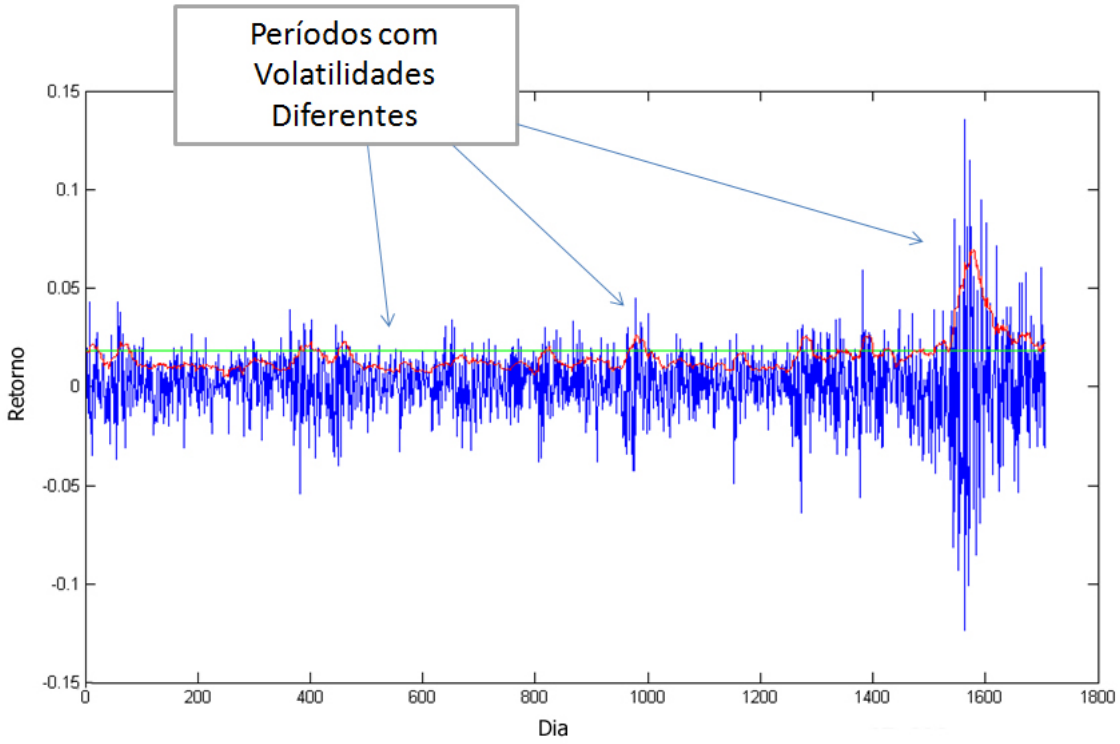

Figura 41: Análise de volatilidade - Mudanças de comportamento

Os momentos em que a curva de desvio padrão instantâneo (s_{inst}) ultrapassa em amplitude a curva de desvio padrão total (s_{total}) são os momentos em que a série torna-se mais instável ou volátil, de acordo com a regra descrita pela equação (4.8). No entanto, nota-se que a própria amplitude do sinal também serve como indicador visual da mudança de comportamento da série, na maioria dos casos.

$$volat(r_t) = \begin{cases} +1\ (alta)\ se\ s_{inst}(r_t) \geq s_{total} \\ -1\ (baixa)\ se\ s_{inst}(r_t) < s_{total} \end{cases} \tag{4.8}$$

Sabe-se que, em uma série com distribuição normal, aproximadamente 68% dos dados se encontram no intervalo entre a média e o valor de um desvio padrão (regra 68-95-99,7). No caso da série estudada, cerca de um terço dos dados foram classificados como

representantes de períodos de "alta volatilidade" e dois terços (aproximadamente 68%) como "baixa volatilidade", indicando que a série realmente se aproxima de uma coleção de variáveis aleatórias com distribuição normal.

Segundo Tsay (2002), algumas características básicas sobre a volatilidade em séries de retorno são de grande importância para o desenvolvimento de modelos que a levem em consideração, e todas elas podem ser comprovadas através de análise da Figura 41:

- Existem *clusters* (blocos) de volatilidade, ou seja, a volatilidade muda de tempos em tempos, apresentando-se alta por certos períodos e baixa em outros;
- A volatilidade evolui continuamente no tempo, de maneira que a probabilidade de ocorrência de saltos de volatilidade é muito remota;
- A medida de volatilidade nunca diverge para o infinito, mas varia sempre em um intervalo fixo.

4.7.2 Construção da Nova Arquitetura

Com a finalidade de aproveitar separadamente as características intrínsecas aos períodos de alta e baixa volatilidade, foi desenvolvida uma nova arquitetura para a previsão, na qual são utilizados dois modelos hierárquicos HNM ao invés de apenas um, como utilizado anteriormente. O funcionamento da nova arquitetura (HNM-V, modelo neural hierárquico com volatilidade), está ilustrado na Figura 42 abaixo:

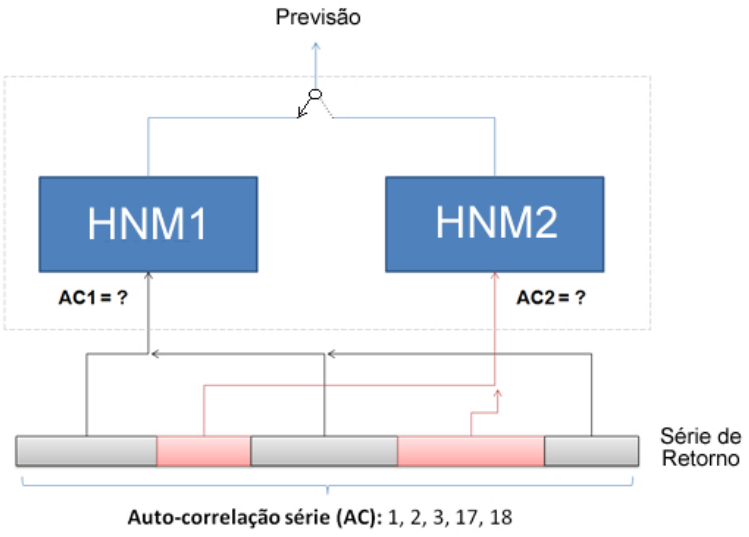

Figura 42: Nova Arquitetura HNM-V: Dois modelos hierárquicos

Na Figura 42, a série de retorno é representada na parte inferior por um retângulo dividido em setores cinzas (períodos de baixa volatilidade) e vermelhos (alta volatilidade), tratados separadamente pelos modelos hierárquicos representados por retângulos azuis identificados por HNM1 e HNM2. Com isso, atribui-se a um dos modelos a função de aprender apenas a dinâmica dos períodos de estabilidade, enquanto o outro se ocupa exclusivamente dos períodos de instabilidade.

A nova arquitetura funciona da seguinte maneira: Antes que um padrão seja fornecido à entrada do modelo, realiza-se o cálculo do desvio padrão instantâneo, utilizando os vinte dias anteriores ao dia "atual" (Equação 4.6). Comparando-se o resultado obtido com o desvio padrão total da série, classifica-se o dia como pertencente a um período de alta ou baixa volatilidade. O modelo monta, então, o padrão de entrada correto, de acordo com as autocorrelações, e o direciona para a entrada da rede específica, exclusiva para tratamento de dados do tipo encontrado na classificação (HNM1 ou HNM2). A partir daí, os processos são idênticos aos descritos nos experimentos anteriores, sendo que a única diferença é que o treinamento ocorre paralelamente em duas redes, disponibilizando dois modelos - especializados em situações opostas - para a previsão.

Nos experimentos realizados anteriormente, a informação relacionada à volatilidade era explícita no padrão de entrada, mas tratada por uma única rede. Na arquitetura HNM-V, cada uma das redes é ajustada especificamente para a previsão de dados provenientes de períodos instáveis ou estáveis. Esta divisão permite que o usuário seja capaz de ajustar individualmente os parâmetros tanto da SVM (ε, C) quanto da rede SOM (tamanho do mapa, função de vizinhança), tirando proveito das particularidades dos dois modos de volatilidade e buscando resultados ótimos para cada um deles separadamente. Em resumo, aplica-se o velho e conhecido conceito "dividir para conquistar".

Além disso, é importante notar o fato de que os valores de autocorrelação calculados para a série como um todo não são necessariamente os mesmos para cada uma das séries obtidas com a sua divisão em períodos de volatilidade distinta. Por este motivo, torna-se necessária a realização de novos cálculos para definição dos valores de autocorrelação, que devem ser obtidos separadamente para as séries de alta e de baixa volatilidade. Utilizando-se a equação (4.4) para cada uma delas, obtém-se os novos valores de autocorrelação que, expressos em correlogramas, podem ser vistos na Figura 43 abaixo:

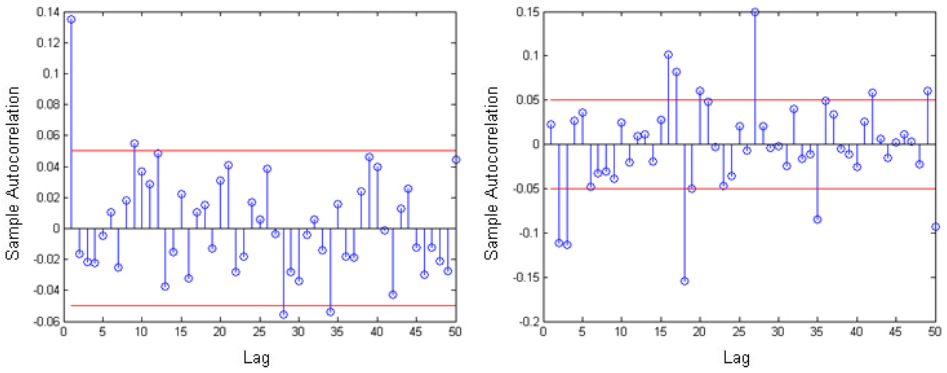

Figura 43: Autocorrelação: Baixa volatilidade / Alta volatilidade

Observando os resultados obtidos, encontramos os valores de autocorrelação que se mostraram mais significativos (acima de 5%) e somos capazes de identificar os *lags* que compõem os padrões de entrada para cada um dos modelos hierárquicos:

Baixa volatilidade (HNM1): *lags* 1, 9, 12, 28 e 34;

Alta volatilidade (HNM2): *lags* 2, 3, 16, 18, 27.

No caso desta nova arquitetura, não existe a necessidade de se colocar informação sobre a volatilidade no padrão de entrada, como nos testes realizados com apenas um modelo hierárquico. Aqui, esta informação é processada inicialmente e apenas os dados históricos de retorno são passados para um modelo ou outro, dependendo do comportamento atual da série, formando um padrão de entrada com cinco componentes.

4.7.3 Resultados Obtidos

Para os testes, foram utilizados novamente os dez períodos escolhidos para os experimentos anteriores, sendo cinco deles específicos para cada modo de volatilidade. Muitos testes foram realizados, com diferentes parâmetros na máquina de vetor de suporte, mapas auto-organizáveis de variadas dimensões, diferentes funções de vizinhança, e o resultado do melhor deles está representado na tabela 6 ao lado dos resultados alcançados anteriormente pelos demais modelos, para efeito de comparação. Os erros se encontram expressos novamente em termos de erro percentual absoluto médio (MAPE).

Tabela 6: Resultados: Divisão por volatilidade

Experimento	**MLP**	**SVM**	**HNM**	**HNM-V**
1B	1,5541	**1,1785**	4,1581	1,2434
2B	2,9294	2,4037	**0,9902**	3,2745
3B	4,3077	2,6339	5,8239	**1,6491**
4B	1,7597	**1,1738**	1,9167	1,4496
5B	4,1239	2,0040	2,5711	**1,8358**
1A	6,1807	5,8912	7,1402	**4,0869**
2A	9,1115	8,1148	**5,0196**	5,3138
3A	3,5636	2,8835	**2,1515**	3,7272
4A	10,6918	10,5901	6,2069	**5,4878**
5A	2,3696	2,8112	**1,9342**	4,9700
Erro médio	**4,6592**	**3,9685**	**3,7912**	**3,3038**
Desvio Padrão	3,0988	3,1786	2,1554	1,6652

Os parâmetros utilizados pelos modelos responsáveis pela previsão em baixa e alta volatilidade estão descritos nas tabelas 7 e 8, respectivamente. Como podemos observar, os parâmetros utilizados na rede SOM são diferentes de um modelo para o outro (dimensões do mapa, raio de vizinhança), resultando em uma previsão altamente especializada, com capacidade de generalização mais eficiente.

Tabela 7: Parâmetros: Modelo Hierárquico - Baixa volatilidade

SOM	
Dimensões	18 x 18
Entrada	5 entradas
Raio da Vizinhança	2
Treinamento (*Coarse-Mapping*)	
Épocas	400
Taxa de Aprendizagem Inicial	0,5
Raio da Vizinhança Inicial	18
Treinamento (*Fine-Tuning*)	
Épocas	1000
Taxa de Aprendizagem Inicial	0,01
Raio da Vizinhança Inicial	1
SVM	
Tipo	ε-SVR
Função de *Kernel*	Função de base radial
C	1000
ε	0,005

Tabela 8: Parâmetros: Modelo Hierárquico - Alta volatilidade

SOM	
Dimensões	50 x 50
Entrada	5 entradas
Raio da Vizinhança	3
Treinamento (*Coarse-Mapping*)	
Épocas	400
Taxa de Aprendizagem Inicial	0,5
Raio da Vizinhança Inicial	50
Treinamento (*Fine-Tuning*)	
Épocas	1000
Taxa de Aprendizagem Inicial	0,01
Raio da Vizinhança Inicial	1
SVM	
Tipo	ε-SVR
Função de *Kernel*	Função de base radial
C	1000
ε	0,005

É interessante notar que o mapa auto-organizável responsável pelos dados de alta volatilidade é muito maior do que o de baixa, levando à conclusão que a formação de um contexto para estes dados é mais complicada e necessita de um mapa com maior resolução.

As figuras de 44 a 53 mostram um comparativo visual entre os resultados obtidos pelo modelo hierárquico isolado (HNM, em vermelho) e pela arquitetura que utiliza os dois modelos paralelamente, explorando a volatilidade (HNM-V, em verde).

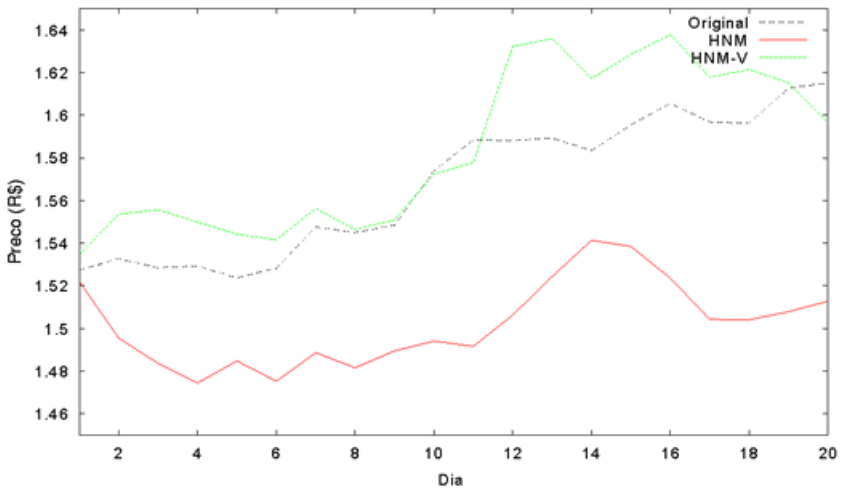

Figura 44: Previsão c/ volatilidade: Período 1, Baixa volatilidade (1B)

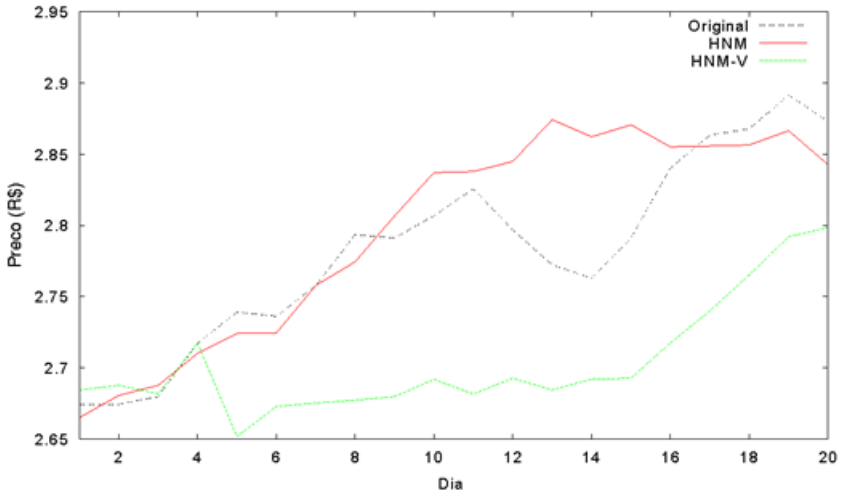

Figura 45: Previsão c/ volatilidade: Período 2, Baixa volatilidade (2B)

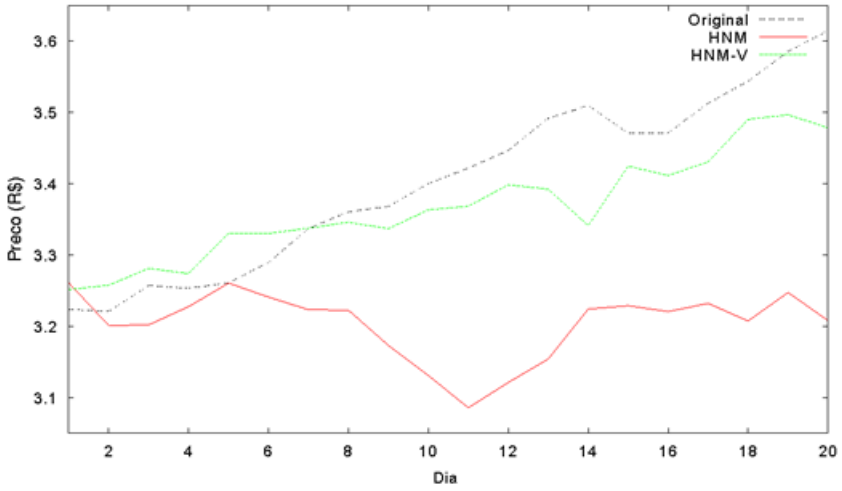

Figura 46: Previsão c/ volatilidade: Período 3, Baixa volatilidade (3B)

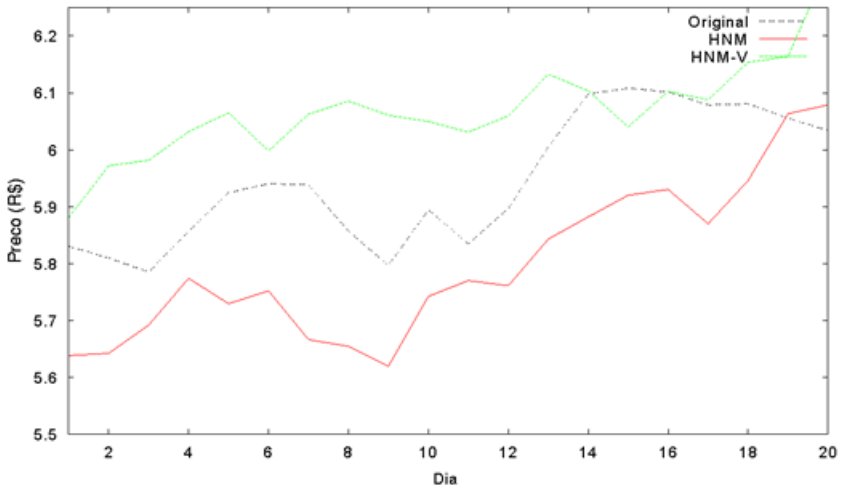

Figura 47: Previsão c/ volatilidade: Período 4, Baixa volatilidade (4B)

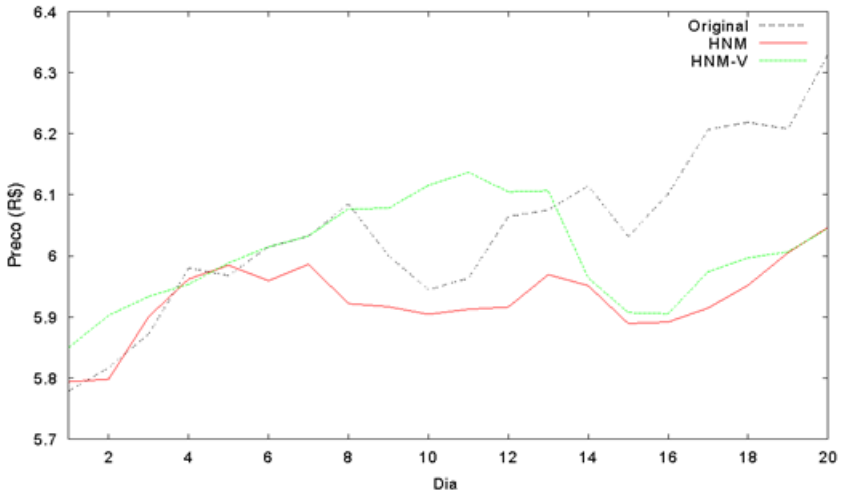

Figura 48: Previsão c/ volatilidade: Período 5, Baixa volatilidade (5B)

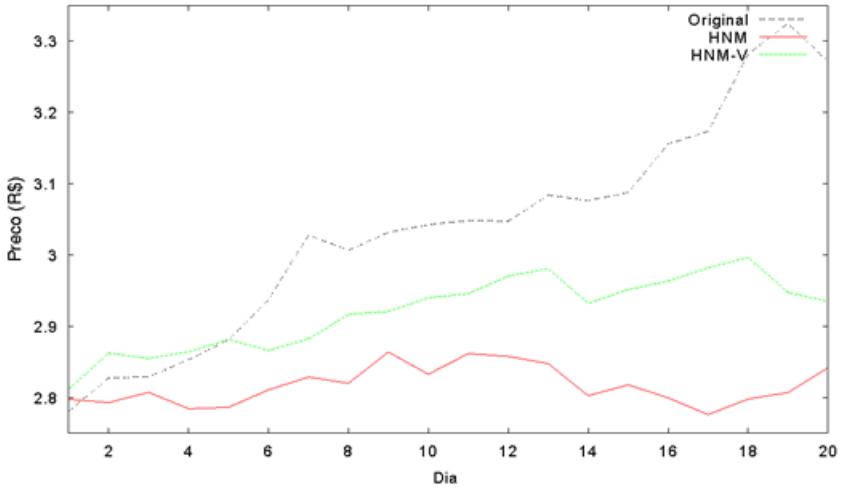

Figura 49: Previsão c/ volatilidade: Período 1, Alta volatilidade (1A)

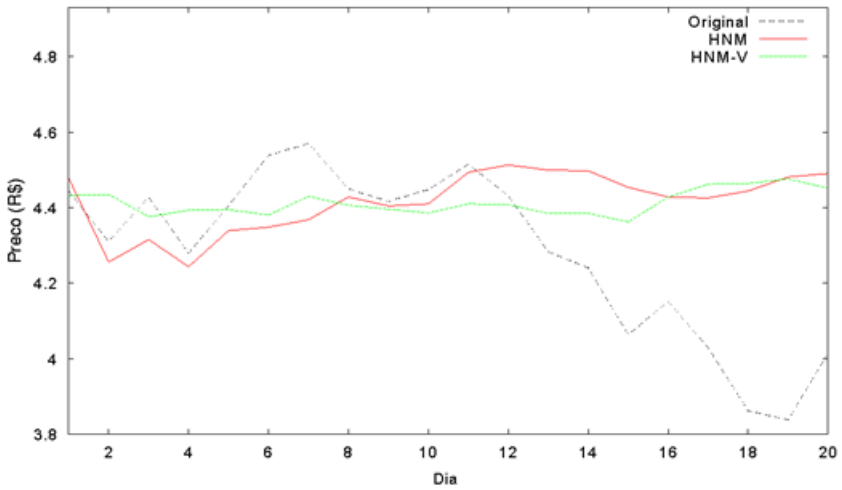

Figura 50: Previsão c/ volatilidade: Período 2, Alta volatilidade (2A)

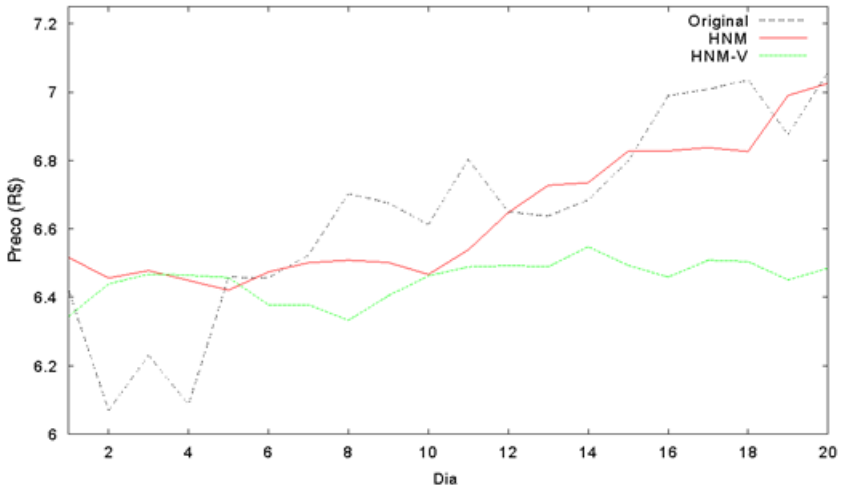

Figura 51: Previsão c/ volatilidade: Período 3, Alta volatilidade (3A)

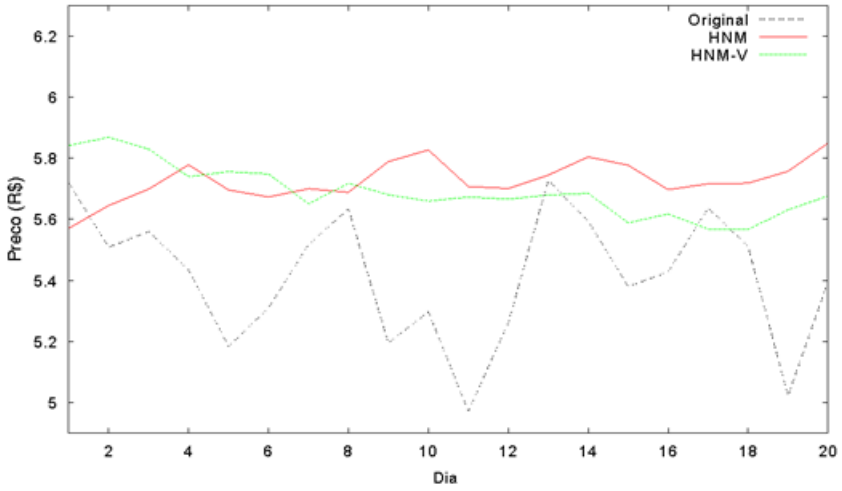

Figura 52: Previsão c/ volatilidade: Período 4, Alta volatilidade (4A)

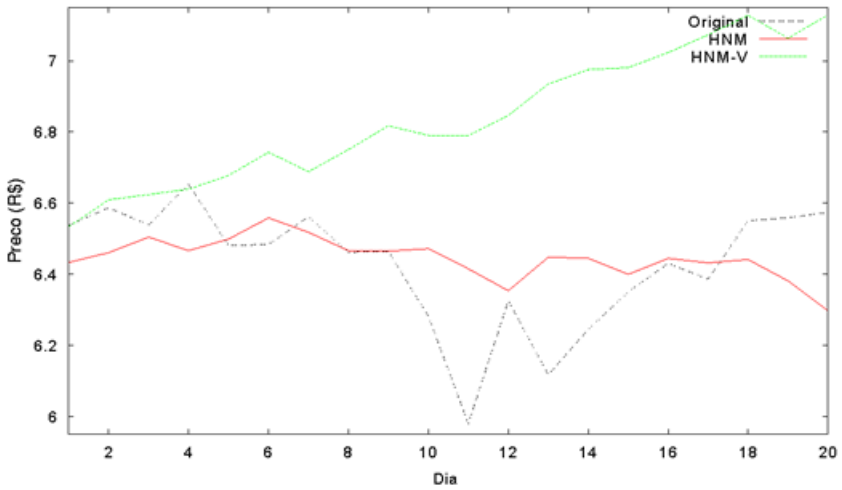

Figura 53: Previsão c/ volatilidade: Período 5, Alta volatilidade (5A)

Através da média aritmética dos erros obtidos nos dez experimentos, comprova-se que a arquitetura formada por dois modelos hierárquicos apresenta os melhores resultados na previsão da série estudada. Os resultados obtidos pelo HNM-V foram superiores a todos os demais - incluindo o modelo HNM individual - em quatro dos experimentos (3B, 5B, 1A e 4A) e apresentou um erro desproporcional em apenas um deles (5A). Em relação apenas ao modelo proposto anteriormente, o HNM, a nova arquitetura foi superior em seis dos testes, além de apresentar um erro médio substancialmente menor. Resumidamente, podemos considerar que o modelo HNM-V manteve a qualidade de previsão do modelo HNM para períodos de alta volatilidade e apresentou sensível melhora em períodos de baixa volatilidade.

Em termos percentuais, a previsão utilizando a arquitetura HNM-V alcançou resultados melhores que os da rede MLP em 29%, à máquina de vetor de suporte pura em 16,75% e ao modelo hierárquico proposto inicialmente em aproximadamente 12,7%. Além disso, os erros apresentados pelo modelo HNM-V mostraram-se menos dispersos, variando dentro de um intervalo mais estreito que os demais, o que resulta em uma maior confiabilidade e indica menores chances para a ocorrência de um erro muito alto. Calculando-se o desvio padrão dos resultados obtidos nos dez experimentos para cada um dos modelos obtemos a tabela 9, abaixo:

Tabela 9: Desvio Padrão - Dispersão das medidas de erro

Modelo	**MLP**	**SVM**	**HNM**	**HNM-V**
Desvio Padrão	3,0988	3,1786	2,1554	**1,6652**

Outro fator de grande importância é a análise do aspecto gráfico das previsões, onde percebe-se novamente que a curva que representa os valores previstos acompanha com certa fidelidade a curva real. Tanto o modelo HNM-V quanto o HNM testado anteriormente se mostraram sensíveis a pequenas oscilações, perdendo-se eventualmente mas apresentando bons resultados na maioria dos experimentos. É importante salientar também que as duas arquiteturas se mostraram superiores aos modelos estabelecidos como parâmetros para comparação (MLP e SVM), tanto no que diz respeito ao erro percentual absoluto médio, quanto à fidelidade no seguimento da tendência da curva. Daí, comprova-se que a codificação e manipulação eficiente de informações de contexto gera benefícios para o modelo, resultando em previsões de maior qualidade.

Finalmente, o modelo HNM-V pode ser considerado a melhor alternativa para a previsão na série de fundo de investimentos, devido aos números absolutos obtidos para as

medidas de erro - menores e menos dispersos - e pela manutenção da eficiência, independentemente do comportamento da série e oscilações do mercado. Seu desempenho superior é justificado pelo alto nível de especialização atingido para cada um dos modos de volatilidade (alta e baixa) através da divisão da série, realizada de acordo com seu comportamento recente. A divisão do espaço de entrada viabiliza a criação de mapas de características e contextos separados para dados com características distintas, formados através de treinamento específico, apenas com informações relevantes para cada modo de volatilidade, e, por isso mesmo, mais significativos para a previsão.

5 Conclusão

5.1 Discussão dos Resultados

O desenvolvimento de modelos para previsão em séries temporais financeiras é ainda um problema de difícil solução, especialmente devido às características deste tipo de série, discutidas nos capítulos 2 e 4. Este estudo apresenta um modelo que representa uma abordagem original no ataque a este tipo de problema, onde uma estrutura hierárquica é disposta de maneira a construir um contexto para os dados, utilizando duas estruturas amplamente divulgadas em meios de pesquisa: o mapa auto-organizável e a máquina de vetor de suporte.

Estruturas hierárquicas compostas por um estágio com aprendizagem supervisionada sobre outro com aprendizagem não-supervisionada já haviam sido empregados na previsão de séries temporais financeiras, mas com objetivos distintos. Na maioria deles, a função do primeiro estágio - geralmente uma rede SOM - é segmentar os dados em diferentes regiões, que passam a ser tratadas por redes especialistas, representadas quase sempre por máquinas de vetor de suporte (HSU et al., 2009) (TAY; CAO, 2001b). No caso deste trabalho, a rede SOM é disposta de maneira a criar contexto para os dados de entrada, transformando estas informações em um mapa de características que aproxima o espaço de entrada e destaca suas propriedades estatísticas mais importantes. É através deste mapa de características que a SVM aprimora seu desempenho, apresentando resultados mais eficientes.

No decorrer da análise, constatou-se também que os dados da série de fundo de investimentos se apresentam muitas vezes em agrupamentos bem separados, ou *clusters*, que se revezam ciclicamente no tempo e são caracterizados por comportamentos distintos, onde a série demonstra maior ou menor volatilidade. Por este motivo, desenvolveu-se uma nova arquitetura composta por dois modelos hierárquicos especializados em comportamentos distintos e uma fase inicial, responsável pela segmentação da série em períodos de alta ou baixa volatilidade, baseada no cálculo do desvio padrão de seu histórico recente.

Finalmente, combinando a utilização de estruturas hierárquicas para construção de contexto para os dados (CARPINTEIRO et al., 2007) (CARPINTEIRO; BARROW, 1996) e a segmentação do espaço de entrada para utilização de modelos especialistas (HSU et al., 2009) (TAY; CAO, 2001b), criou-se uma nova arquitetura, chamada HNM-V, ou modelo neural hierárquico com volatilidade.

A proposição de um modelo de previsão para o auxílio de profissionais do sistema financeiro, ou mesmo para investidores de ocasião, alcançou resultados satisfatórios e dentro dos objetivos iniciais. Os resultados obtidos pelo modelo hierárquico simples (HNM) e, especialmente, pelo modelo hierárquico com suporte a volatilidade (HNM-V) se mostraram superiores aos de modelos neurais estabelecidos nos meios comercial e acadêmico, representados pelo perceptron de múltiplas camadas (MLP) e pela máquina de vetor de suporte (SVM), e justificam - ou mesmo incentivam - a sua utilização para a finalidade proposta.

Ambas as arquiteturas se mostraram superiores aos modelos estabelecidos como parâmetros de comparação, em relação tanto ao erro percentual absoluto médio obtido nos experimentos quanto, também, à captura da dinâmica da série, constatada pela maior fidelidade no seguimento da tendência da curva, comprovando que a manipulação eficiente de informações de contexto gera benefícios para o modelo e resulta em previsões de maior qualidade.

Com base nos resultados obtidos, o modelo HNM-V pode ser finalmente considerado a melhor alternativa para a previsão na série de fundo de investimentos, devido aos valores de erro menores e menos dispersos e pela manutenção da eficiência, independentemente da volatilidade no período de interesse. Seu desempenho, superior ao modelo HNM individual, pode ser justificado pelo nível de especialização alcançado pelos dois modelos hierárquicos que o compõem. A divisão da série possibilita a criação de contextos separados para dados com características distintas, mais significativos exatamente por serem formados a partir somente de informação relevante para o respectivo modo de volatilidade.

5.2 Considerações Finais e Trabalhos Futuros

Apesar dos bons resultados alcançados, alguns pontos negativos do modelo desenvolvido devem ser levados em consideração no processo de escolha de um previsor por um profissional da área. Os modelos hierárquicos (HNM e HNM-V) apresentam em sua estrutura uma complexidade consideravelmente maior que os demais modelos testados, o que acaba resultando em uma utilização também superior de recursos da máquina onde

estão sendo executados. O software desenvolvido pode gerar arquivos temporários bastante extensos, de acordo com a dimensão do mapa definida pelo usuário, e consumir uma quantidade razoável de tempo de CPU.

Observa-se ainda, que o tempo necessário para seu treinamento é bastante superior ao dos demais modelos, chegando a ser dez vezes maior que o do perceptron de multiplas camadas, para os resultados expressos nesta dissertação. Mesmo assim, os tempos gastos no treinamento dos modelos que utilizam estrutura hierárquica mantêm-se na ordem de minutos e são justificados pelos resultados de melhor qualidade, lembrando que esta característica não é particularmente prejudicial na sua aplicação cotidiana, uma vez que o horizonte de previsão é da ordem de dias. A diminuição do tempo de execução e otimização do software utilizado não foram prioridades durante a elaboração do modelo hierárquico e, portanto, seu estudo detalhado visando a otimização e diminuição da complexidade é uma sugestão para pesquisas futuras.

Outra preocupação recorrente no estudo das máquinas de vetor de suporte é a falta de um mecanismo padrão para a escolha dos parâmetros livres. Durante os estudos, os parâmetros ε e C foram escolhidos empiricamente, tornando a experiência de ajuste da ε-SVR uma tarefa cansativa, sujeita à imprecisão do operador humano e com considerável exigência de tempo. Segundo Kaastra e Boyd (1996), a própria previsão pode ser comprometida muitas vezes pela escolha inadequada dos parâmetros do previsor. Por este motivo, o desenvolvimento de métodos para a escolha dos parâmetros livres da máquina de vetor de suporte permanece um campo de pesquisa em aberto e serve também como sugestão para trabalhos futuros.

Finalmente, a obtenção de bons resultados utilizando o modelo hierárquico desenvolvido neste estudo adiciona uma alternativa promissora ao conjunto de ferramentas existentes para previsão em séries temporais financeiras. São muitas as possibilidades para o seu aperfeiçoamento, através de mudanças no código, ajuste e inclusão de novos parâmetros livres, divisão do espaço de entrada em uma quantidade maior de modos de volatilidade, adição de novos valores ao padrão de entrada, e muitos outros aspectos que podem vir a trazer benefícios para o previsor, além de um estudo comparativo com métodos estatísticos não-lineares (como o GARCH, por exemplo), que não foi realizado neste trabalho. Assim, o campo fica aberto à espera de novas ideias e da criatividade de novos pesquisadores.

Referências

ANDERSON, J. *Cognitive Psychology and Its Implications*. Third edition. New York: W. H. Freeman, 1990.

ARMANO, G.; MARCHESI, M.; MURRU, A. A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*, v. 170, n. 1, p. 3 – 33, 2005. Computational Intelligence in Economics and Finance.

ASIMOV, I. *I, Robot*. [S.l.]: Garden City, N.Y., Doubleday, 1963.

BB DTVM. *Regulamento do BB Ações IBrX Indexado*. [S.l.], 2010.

BENNETT, K. P.; CAMPBELL, C. Support vector machines: hype or hallelujah? *SIGKDD Explorations Newsletter*, ACM, New York, NY, USA, v. 2, n. 2, p. 1–13, 2000.

BOWERMAN, B. L.; O'CONNELL, R. T.; KOEHLER, A. B. *Forecasting, Time Series and Regression: An Applied Approach*. 4th edition. ed. [S.l.]: Thomson Brooks/Cole, 2005.

BOX, G. E. P.; JENKINS, G. M. *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day, 1976.

CAMPBELL, J. Y.; LO, A. W.; MACKINLAY, A. C. *The Econometrics of Financial Markets*. [S.l.]: Princeton Univesity Press, 1996.

CANU, S. et al. *SVM and Kernel Methods Matlab Toolbox*. 2005. Perception Systèmes et Information, INSA de Rouen, Rouen, France. Disponível em: `http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html`.

CAO, L. Support vector machines experts for time series forecasting. *Elsevier Neurocomputing*, n. 51, p. 321–339, Fevereiro 2002.

CAO, L. Support vector machines experts for time series forecasting. *Neurocomputing*, v. 51, p. 321 – 339, 2003.

CAO, L.; TAY, F. E. H. Application of support vector machines in financial time series forecasting. *Omega*, v. 29, n. 4, p. 309–317, August 2001.

CAO, L.; TAY, F. E. H. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, v. 14, n. 6, Novembro 2003.

CARPINTEIRO, O. A. et al. Long-term load forecasting via a hierarchical neural model with time integrators. *Electric Power Systems Research*, v. 77, n. 3-4, p. 371 – 378, 2007.

CARPINTEIRO, O. A. S.; BARROW, H. G. A self-organizing map model for sequence classification. *Cognitive Science Research Papers - University of Sussex*, Julho 1996.

CHANG, C.-C.; LIN, C.-J. *LIBSVM: a library for support vector machines*. [S.l.], 2001. Disponível em: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

CLEMENTS, M. P.; FRANSES, P. H.; SWANSON, N. R. Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting*, v. 20, n. 2, p. 169 – 183, 2004.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, p. 273–297, 1995.

DEBOECK, G. J. Financial applications of self-organizing maps. *American Heuristics Electronic Newsletter*, junho 1998.

DRAKE, C.; PALMER, C. Accent structures in music performance. *Music Perception: An Interdisciplinary Journal*, v. 10, n. 3, p. 343–378, 1993.

ENGLE, R. F. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, v. 50, n. 4, p. pp. 987–1007, 1982.

FAN, S.; CHEN, L. Short-term load forecasting based on an adaptive hybrid method. *Power Systems, IEEE Transactions on*, v. 21, n. 1, p. 392 – 401, feb. 2006.

FAN, S.; MAO, C.; CHEN, L. Next-day electricity-price forecasting using a hybrid network. *Generation, Transmission Distribution, IET*, v. 1, n. 1, p. 176 –182, jan. 2007.

FUNDOS de Ações - Banco do Brasil. jul. 2010. Disponível em: <`http://www21.bb.com.br/portalbb/cotaFundos/GFI9,2,001.bbx?tipo=1\&fundo=6`>.

GOOGLE Acadêmico. set. 2010. Disponível em: <`http://scholar.google.com`>.

HAWKINS, J.; BLAKESLEE, S. *On Intelligence*. [S.l.]: Times Books, 2004.

HAYKIN, S. *Redes Neurais: Princípio e Prática*. 2nd. ed. [S.l.]: Bookman, 2001.

HILL, T.; O'CONNOR, M.; REMUS, W. Neural network models for time series forecasts. *Management Science*, v. 42, n. 7, p. 1082–1092, 1996.

HOCHREITER, S.; OBERMAYER, K. Support vector machines for dyadic data. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 18, n. 6, p. 1472–1510, 2006.

HSU, S.-H. et al. A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Systems with Applications*, v. 36, n. 4, p. 7947 – 7951, 2009.

KAASTRA, I.; BOYD, M. Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, v. 10, n. 3, p. 215 – 236, 1996. Financial Applications, Part II.

KIM, K. jae. Financial time series forecasting using support vector machines. *Neurocomputing*, v. 55, n. 1-2, p. 307 – 319, 2003.

KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, v. 78, n. 9, p. 1464 – 1480, 1990.

KOHONEN, T. *Self-Organizing Maps*. Third edition. Berlin: Springer-Verlag, 2001.

LERDAHL, F.; JACKENDOFF, R. *A Generative Theory of Tonal Music*. Cambridge, MA: The MIT Press, 1983.

MAKEY, M.; GLASS, L. Oscillation and chaos in physiological control systems. *Science*, v. 197, n. 287, 1977.

MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; HYNDMAN, R. J. *Forecasting: Methods and Appplications*. 3rd edition. ed. [S.l.]: John Wiley & Sons, Inc., 1998.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943.

MERCER, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, v. 83, n. 559, p. 69–70, 1909.

MüLLER, K. et al. Predicting time series with support vector machines. In: GERSTNER, W. et al. (Ed.). *Artificial Neural Networks - ICANN'97*. [S.l.]: Springer Berlin / Heidelberg, 1997, (Lecture Notes in Computer Science, v. 1327). p. 999–1004.

PANKRATZ, A. *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*. New York: John Wiley, 1983.

PéREZ-CRUZ, F.; BOUSQUET, O. Kernel methods and their potential use in signal processing. *Signal Processing Magazine, IEEE*, v. 21 Issue: 3, p. 57 – 65, May 2004.

ROSENBLATT, M. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, p. 386–408, 1958.

RUMELHART, D. E.; MCCLELLAND, J. L. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.

SMOLA, A. J.; SCHöLKOPF, B. A tutorial on support vector regression. *Statistics and Computing*, Kluwer Academic Publishers, Hingham, MA, USA, v. 14, n. 3, p. 199–222, 2004.

TAY, F. E. H.; CAO, L. Application of support vector machines in financial time series forecasting. *Omega*, v. 29, n. 4, p. 309 – 317, 2001.

TAY, F. E. H.; CAO, L. J. Improved financial time series forecasting by combining support vector machines with self-organizing feature map. *Intell. Data Anal.*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 5, n. 4, p. 339–354, 2001.

TSAY, R. S. *Analysis of Financial Time Series*. 2nd. ed. [S.l.]: Wiley-Interscience, 2002.

VAPNIK, V. N. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

VAPNIK, V. N. *Statistical Learning Theory*. [S.l.]: Wiley-Interscience, 1998.

WEIGEND, A. S.; GERSHENFELD, N. A. *Time Series Prediction: Forecasting the Future and Understanding the Past*. [S.l.]: Addison-Wesley, 1994.

YANG, H.; CHAN, L.; KING, I. Support vector machine regression for volatile stock market prediction. In: YIN, H. et al. (Ed.). *Intelligent Data Engineering and Automated Learning - IDEAL 2002*. [S.l.]: Springer Berlin Heidelberg, 2002, (Lecture Notes in Computer Science, v. 2412). p. 143–152.