

Descoberta de sinais de compra e venda ótimos para análise técnica financeira do mercado de ações usando redes neurais artificiais

**Trabalho de Conclusão de Curso
Engenharia da Computação**

Murilo Rebelo Pontes

Orientador: Adriano Lorena Inácio Oliveira, Phd.

Recife, 28 de maio de 2008

Descoberta de sinais de compra e venda ótimos para análise técnica financeira do mercado de ações usando redes neurais artificiais

Trabalho de Conclusão de Curso Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de

Murilo Rebelo Pontes

Orientador: Prof. Adriano Lorena Inácio Oliveira, Phd.

Recife, maio de 2008

Murilo Rebelo Pontes

**Descoberta de sinais de compra e venda
ótimos para análise técnica financeira do
mercado de ações usando redes neurais
artificiais**

Resumo

Uma das áreas de estudos e aplicações práticas de redes neurais artificiais é previsão de séries temporais. Séries temporais são informações históricas de uma determinada variável, ou seja, são dados que variam com o tempo. São exemplos de séries temporais: séries metrológicas, séries financeiras, sorteios de loterias, e qualquer outra série que se tenha anotado o tempo e o valor uma ou varias variáveis.

No mercado financeiro a principal meta é ter lucros cada vez maiores. Para se obter tais lucros é preciso fazer uma análise técnica e determinar os pontos de compra e venda de cada ação negociada nas bolsas de valores ao redor do mundo. Análise técnica é feita calculando-se dos indicadores técnicos de uma dada ação usando as séries temporais do mesmo, cada indicador possui limiares que geram sinais de compra ou venda. Há diversos indicadores técnicos que são calculados com base nas séries temporais correspondentes às ações.

O sistema proposto neste trabalho tem como meta maximizar os lucros dos investidores, através da utilização de sistemas multiagente inteligentes que utilizam técnicas de redes neurais artificiais para implementar aprendizado de máquina. Algoritmos com Backpropagation, RPROP, Cascade Correlation foram implementados para serem testados neste problema de previsão, também foram usadas algumas implementações de terceiros como libSVM e Weka. O artefato final produzido um agente inteligente que, combinando todos os subsistemas de forma que o progresso de negociação de compra e venda de ações com lucro maximizado, foi totalmente automatizado. Os resultados experimentais indicam que o agente inteligente acerta em 78,75% de suas previsões.

Abstract

One area of study of artificial neural networks is forecasting time series. Time series are historical information of a particular variable. Examples of time series: metrological series, financial series, raffles, lotteries, and any other series that has been noted time and the value one or several variables.

In the financial market the main goal is to have ever-increasing profits. To achieve such profits must be a technical analysis and determine the points of purchase and sell of individual stocks traded on stock exchanges around the world. Review technique is done with an estimated of technical indicators of a particular stock using the time series of it, each indicator has thresholds that generate signals to buy or sell.

The system proposed in this work aims to maximize the profits of investors, through the use of intelligent systems multiagente using techniques of artificial neural networks to implement learning machine. Algorithms with Backpropagation, RPROP, Cascade Correlation were implemented to be tested this problem of forecasting, were also used some implementations of others as libSVM and Weka. The final artifact produced an intelligent agent that combining all the subsystems so that the progress of negotiation for purchase and sell of shares at a profit maximized was completely automated. The experimental results indicate that the intelligent agent hits in 78.75% of its forecast.

Sumário

Índice de Figuras	8
Índice de Tabelas	9
Tabela de Símbolos e Siglas	10
Agradecimentos	11
1 Introdução	12
1.1 Motivação	13
1.2 Objetivos do Trabalho	14
2 Fundamentação teórica	15
2.1 Classificação de funções quanto à linearidade	15
2.2 Interpolação	15
2.3 Redes neurais Artificiais	15
2.3.1 Perceptron	16
2.4 Multilayer Perceptron	17
2.4.1 Algoritmo Backpropagation	18
2.4.2 Problemas não lineares	19
2.4.3 Algoritmo Resilient Backpropagation (RPROP)	19
2.4.4 Algoritmo Cascade Correlation	21
2.5 Support Vector Machine	24
2.6 Agentes Inteligentes	25
2.6.1 Agentes com aprendizado	26
2.6.2 Sistema Multiagente inteligente	26
2.7 Data Mining	27
2.8 Análise técnica Financeira	27
2.8.1 Relative Strength Index	27

	7
2.8.2 Usando RSI por valor	29
2.8.3 Usando RSI por cruzamento de linha central	29
3 Metodologia de Estudo	30
3.1 Obter dados históricos	30
3.2 Ordens de compra e venda usando regra D+1	30
3.3 Calcular <i>RSI</i>	31
3.4 Relação entre <i>RSI</i> e ordens de compra e venda usando regra D+1	31
3.5 Configuração e execução da SVM	32
3.6 Simulação	32
4 Estudo de Caso	34
4.1 Preprocessamento dos dados históricos	34
4.2 Normalização de preços	36
4.3 Sistema multiagente inteligente por ativo	36
4.4 Analisando resultados	37
5 Conclusões e contribuições	39
5.1 Conclusão	39
5.2 Problemas encontrados	40
5.3 Contribuições	40
5.4 Trabalhos Futuros	40

Índice de Figuras

Figura 1.	Estrutura de um Perceptron	16
Figura 2.	Exemplo de problema linearmente separável.	16
Figura 3.	Função sigmóide logística	17
Figura 4.	Exemplo de rede MLP	17
Figura 5.	Pseudocódigo do algoritmo Backpropagation	18
Figura 6.	Função lógica XOR resolvida com Backpropagation	19
Figura 7.	Pseudocódigo do algoritmo Rprop	20
Figura 8.	Rede antes do treinamento	21
Figura 9.	Rede depois do treinamento	22
Figura 10.	Problemas das duas espirais	23
Figura 11.	Saída da rede treinada	23
Figura 12.	Vetores de dados e hiperplanos paralelos	24
Figura 13.	Hiperplano em vermelho maximiza a distância entre vetores dados	24
Figura 14.	Transformação do espaço usando x ao quadrado	25
Figura 15.	Esquemático de um agente inteligente	26
Figura 16.	Agente inteligente adaptado para mercado financeiro	26
Figura 17.	Procedimento para cálculo do RSI	27
Figura 18.	RSI calculado com janelas de 7,14,21 e 28 dias	29
Figura 19.	Passos tomados para realizar o treinamento	31
Figura 20.	Configuração do LIBSVM no WEKA	32
Figura 21.	Como é feito o processo de simulação para uma ação	33
Figura 22.	Visualização dos preços de um ativo em gráfico candlestick	35
Figura 23.	Problema com divisão no preço de um ativo de 1 para 3	35
Figura 24.	Problema com o agrupamento no preço de um ativo de 10 para 1	35
Figura 25.	Processo de normalização atribui ao último valor escala 1.	36
Figura 26.	Matriz de confusão típica do agente inteligente usando SVM.	37
Figura 27.	Agente inteligente negociando no Investidor Virtual	38

Índice de Tabelas

Tabela 1.	Função XOR aplicada a duas variáveis.	19
Tabela 2.	RSI aplicado a um ativo usando janela de 14 dias.	28
Tabela 3.	Limiares de compra e venda para RSI	33
Tabela 4.	Resultado dos experimentos.	37

Tabela de Símbolos e Siglas

BOVESPA – Bolsa de valores de São Paulo

MLP – Multilayer Perceptron

RPROP – Resilient Backpropagation

CC – Cascade Correlation

SVM – Support Vector Machine

PSO – Particle Swarm Optimization

DM – Data Mining

RSI – Relative Strength Index

IFR – Índice de força relativa

LIBSVM – Biblioteca para Support Vector Machine desenvolvida por Chih-Chung Chang e Chih-Jen Lin

WEKA – Ferramenta de mineração de dados desenvolvida pela Universidade de Waikato

Agradecimentos

A minha família e a amigos que sempre estiveram presentes ao meu lado incentivando e suportando esse trabalho.

Ao meu orientador, Prof. Dr. Adriano Lorena Inácio de Oliveira, pela ajuda despendida para a conclusão deste trabalho.

Aos professores Fernando Buarque, Ricardo Massa e Sergio Soares por todas as oportunidades acadêmicas e trabalhos conjuntos desenvolvidos.

Aos professores do DSC pela dedicação e atenção com os alunos e empenho na resolução de problemas da comunidade acadêmica.

Capítulo 1

Introdução

Uma das áreas de estudos de redes neurais artificiais é previsão de séries temporais. Séries temporais são informações históricas de uma determinada variável. São exemplos de séries temporais: séries metrológicas, séries financeiras, sorteios de loterias, e qualquer outra série que se tenha anotado o tempo e o valor uma ou varias variáveis.

A previsão de valores futuros é feita através de rede neural artificial treinada com padrões extraídos da série temporal. Um padrão consiste em uma seqüência de valores passados agrupados dentro de uma janela de tempo. O valor da janela de tempo diz quantos valores passados serão utilizados para a montagem do padrão. A rede neural artificial treinada é enxergada como uma função prediz o valor futuro a partir de valores históricos.

Em série financeiras são formados os preços de abertura, máximo, mínimo, fechamento e volume negociado de ações de uma determinada empresa de capital aberto. Uma ação é a menor parte de uma empresa de capital aberto. Comprar uma ação significa comprar um pedaço de uma empresa. Preço de abertura é aquele praticado no inicio do pregão na bolsa de valores e preço de fechamento é o último preço da ação no pregão. Pregão é tempo no dia que se podem fazer negócios na bolsa de valores. Bolsa de valores é o local onde pode se comprar e vender ações de empresas de capital aberto que estão ali cadastradas.

No mercado financeiro a principal meta é ter lucros cada vez maiores. Para se obter tais lucros é preciso fazer uma análise técnica e determinar os pontos de compra e venda de cada ação negociada nas bolsas de valores ao redor do mundo. Análise técnica é feita calculando-se dos indicadores técnicos de uma dada ação usando as series temporais do mesmo, cada indicador possui limiares que geram sinais de compra ou venda. Indicadores técnicos são cálculos estatísticos que têm como objetivo destacar características (tendência, divergência, índice de

força relativa, media, variância, desvio padrão) da série com a finalidade de facilitar a visualização e análise.

Em análise técnica é comum o uso de três indicadores em média, os indicadores podem ser todos diferentes ou um único indicador calculado com diversos períodos de tempo diferentes. Os especialistas [16] recomendam aos investidores que praticam análise técnica que escolham um indicador técnico e tornem-se especialista nele. Existem vários indicadores técnicos, o índice de força relativa é um dos mais populares [15] [16].

Neste trabalho é utilizado o indicador técnico índice de força relativa [2] com diversas janelas de tempo como entrada para um sistema de multiagente inteligente que faz análise de múltiplas combinações de limiares de compra e venda em busca da configuração que proporcione o melhor desempenho financeiro para o investidor.

1.1 Motivação

O processo de estabilização da economia no Brasil reduziu drasticamente a taxa de juros no país. Os investimentos de renda fixa que em 1994 apresentavam taxas de rendimentos de 55% ao mês para taxas inferiores a 1% ao mês em 2008 [17].

Neste cenário não é mais possível ao investidor aumentar seu patrimônio através de renda fixa. A renda fixa atualmente apenas corrige o valor do investimento da desvalorização causada pela inflação. Portanto o investidor que necessita aumentar seu patrimônio precisa migrar do investimento de renda fixa para renda variável. A maior dificuldade para realização da migração é aumento substancial do risco, visto que o investimento de renda fixa é de baixo risco e o investimento de renda variável é de alto risco. Porém quanto maior o risco maior será o retorno, caso o investimento dê certo.

Visando obter ganhos com renda variável [1] foi experimentado o uso de redes neurais artificiais para a tomada de decisão de compra e venda em bolsa de valores, os resultados foram bastante promissores, pois a taxa de rendimento obtida pela simulação de investimento utilizando as decisões tomadas pela rede neural artificial foi superior a taxa de valorização índice oficial da bolsa de valores para o mesmo período de tempo.

1.2 Objetivos do Trabalho

Os objetivos específicos deste trabalho são:

- Maximizar os lucros na negociação de ações na BOVESPA através do reconhecimento de padrões ótimos de compra e venda.
- Desenvolver um sistema automático de investimento para testar em simuladores de bolsa de valores tais como: <http://web.infomoney.com.br/investimentos/acoes/emacao/>, <http://folhainvest.folha.com.br/>, <http://www.investidorvirtual.com/>

Capítulo 2

Fundamentação teórica

Neste capítulo são apresentados os algoritmos e técnicas necessários para a construção do sistema automático de investimentos e para o entendimento e realização dos experimentos deste trabalho.

2.1 Classificação de funções quanto à linearidade

Uma função $f(x)$ é dita linear se satisfaz as propriedades:

- Aditiva: $f(x + y) = f(x) + f(y)$
- Homogeneidade: $f(\alpha x) = \alpha f(x)$

Caso contrário é não linear. A maioria dos problemas do mundo real é inerentemente não linear.

2.2 Interpolação

Interpolação é uma técnica matemática que faz a aproximação de funções para um determinado conjunto de pontos. Uma das maiores aplicações de interpolação é a reconstrução aproximada de uma função complexa ou desconhecida por meio de uma função mais simples.

2.3 Redes neurais Artificiais

As redes neurais artificiais são um tipo de interpolador não linear. A unidade básica de processamento de uma rede neural artificial é o perceptron. Associação de vários perceptrons em grafos forma uma rede neural artificial. Existem vários tipos de redes neurais artificiais, sendo o mais popular o tipo Multi-Layer Perceptron.

2.3.1 Perceptron

O neurônio artificial [3] é um modelo matemático que simula um neurônio biológico. A Figura 1 ilustra o Perceptron, uma estrutura composta por várias entradas, um bloco de soma e um bloco função de ativação.

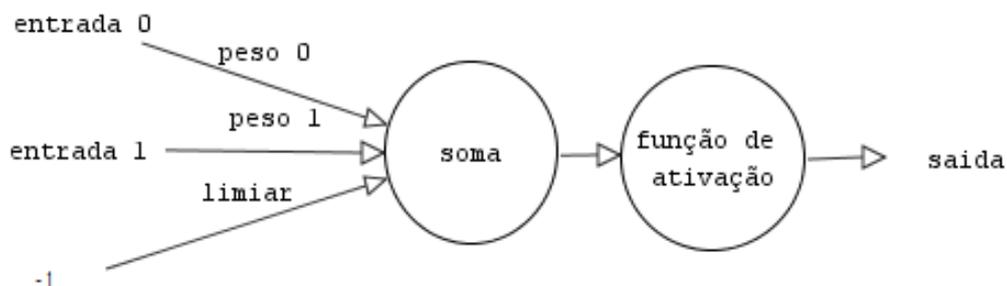


Figura 1. Estrutura de um Perceptron

A cada entrada é atribuído um peso aleatório (na inicialização do treinamento), o somador faz o produto da entrada pelo peso e acumula esse valor, a função de ativação recebe o valor acumulado pelo somador e retorna um hiperplano como saída. Com apenas um hiperplano só é possível resolver problemas linearmente separáveis, na Figura 2 temos um exemplo de problema linearmente separável.

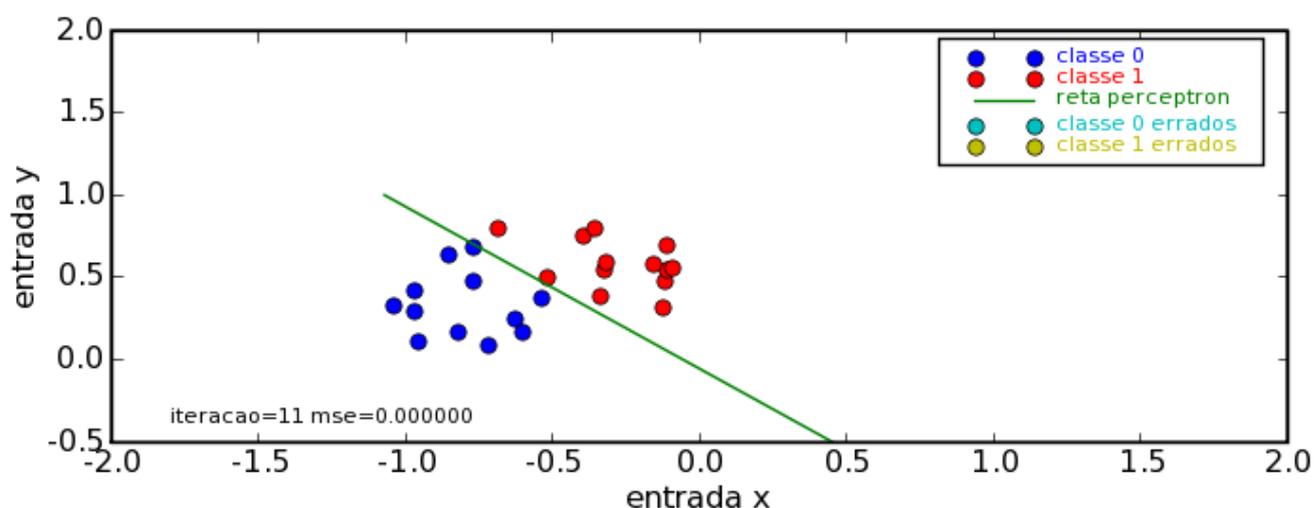


Figura 2. Exemplo de problema linearmente separável.

A função de ativação do perceptron é uma função em formato de S que têm como objetivo simular os limiares de saturação do neurônio biológico. Na Figura 3 podemos ver a função sigmóide logística, utilizada perceptron quando colocado em uma rede de perceptrons com múltiplas camadas, caso o perceptron não faça parte de uma rede de perceptrons é utilizada a função limiar ou degrau unitário, que são funções descontínuas.

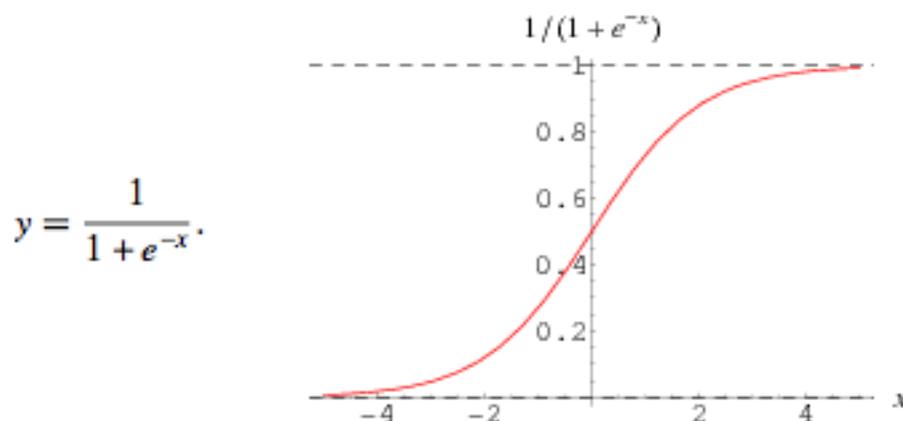


Figura 3. Função sigmóide logística

2.4 Multilayer Perceptron

É uma estrutura composta por vários Perceptrons dispostos em camadas [3], entre as camadas existem conexões que ligam Perceptrons de uma camada com outra. A associação de vários Perceptrons em uma camada permite resolver problemas não linearmente separáveis. Foi provado que redes MLP *feedforward* [10] usando uma camada escondida podem aproximar qualquer função contínua e que redes MLP com duas camadas [11] com pesos usando números reais, podem aproximar qualquer função matemática, tendo assim o mesmo poder da máquina universal de Turing. Na Figura 4 temos um exemplo de uma rede MLP.

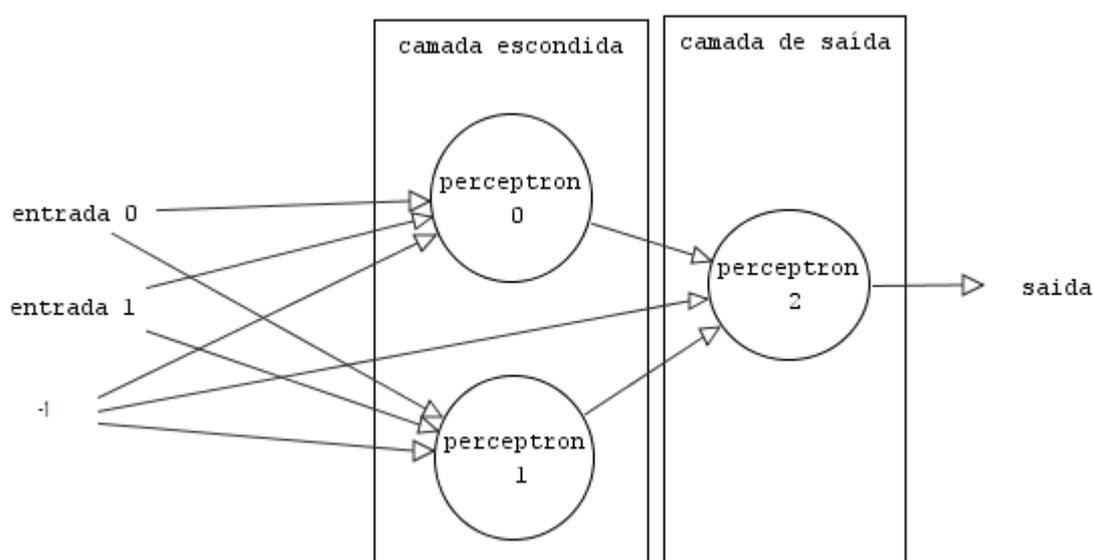


Figura 4. Exemplo de rede MLP

2.4.1 Algoritmo Backpropagation

Nas redes MLP o erro é conhecido apenas nos Perceptrons de saída sendo desconhecido nas camadas internas. As redes MLP passaram quase uma década sem avanços, pela falta de um algoritmo que resolvesse esse problema. Então o algoritmo de Backpropagation [3] é descoberto, inicialmente é atribuído um peso aleatório a cada conexão na rede, tem a rede é ativada no sentido *feedforward*, com os valores das saídas é calculado o erro, então a rede é ativada no sentido *backforward*, com o erro conhecido em todas as camadas é realizado o processo de descida de gradiente a fim de minimizar o erro quadrático médio da saída. O pseudocódigo do algoritmo é mostrado na Figura 5.

$$E_p = \sum_j \frac{(t_{pj} - a_{pj})^2}{2}$$

$$\Delta_p w_{i \rightarrow j} = -\eta \frac{\partial E_p}{\partial w_{i \rightarrow j}}$$

$$\frac{\partial E_p}{\partial w_{i \rightarrow j}} = \frac{\partial E_p}{\partial I_{pj}} \frac{\partial I_{pj}}{\partial w_{i \rightarrow j}}$$

$$\frac{\partial I_{pj}}{\partial w_{i \rightarrow j}} = \frac{\partial}{\partial w_{i \rightarrow j}} \sum_k w_{k \rightarrow j} a_{pk} = a_{pi}$$

$$\delta_{pj} = -\frac{\partial E_p}{\partial I_{pj}}$$

$$\Delta_p w_{i \rightarrow j} = -\eta \delta_{pj} a_{pi}$$

$$\delta_{pj} = -\frac{\partial E_p}{\partial I_{pj}} = -\frac{\partial E_p}{\partial a_{pj}} \frac{\partial a_{pj}}{\partial I_{pj}}$$

$$\frac{\partial a_{pj}}{\partial I_{pj}} = f'_j(I_{pj})$$

Para unidades de saída

$$\frac{\partial E_p}{\partial a_{pj}} = -(t_{pj} - a_{pj})$$

$$\delta_{pj} = (t_{pj} - a_{pj}) f'_j(I_{pj})$$

Para unidades escondidas

$$\sum_k \frac{\partial E_p}{\partial I_{pk}} \frac{\partial I_{pk}}{\partial a_{pj}} = \sum_k \frac{\partial E_p}{\partial I_{pk}} \frac{\partial}{\partial a_{pj}} \sum_i w_{i \rightarrow k} a_{pi} = \sum_k \frac{\partial E_p}{\partial I_{pk}} w_{j \rightarrow k} = -\sum_k \delta_{pk} w_{j \rightarrow k}$$

$$\delta_{pj} = \left(\sum_k \delta_{pk} w_{j \rightarrow k} \right) f'_j(I_{pj})$$

Figura 5. Pseudocódigo do algoritmo Backpropagation

Este algoritmo é executado a cada ativação de um padrão durante o treinamento da rede MLP, ocasionando em um grande custo computacional em comparação com outros algoritmos que são executados uma vez depois que todos os padrões são ativados na fase de treinamento.

2.4.2 Problemas não lineares

Os problemas que não podem ser resolvidos com uma única reta são ditos não lineares. Um exemplo de problema não é a função lógica XOR, na Tabela 1 vemos temos um exemplo da função XOR aplicada a duas variáveis.

Tabela 1. Função XOR aplicada a duas variáveis.

A	b	A xor b
0	0	0
0	1	1
1	0	1
1	1	0

Na Figura 6 vemos este problema resolvido pelo algoritmo de Backpropagation com uma MLP usando 2 perceptrons na camada escondida e 1 perceptron na camada de saída, a conversão de valores inteiro para lógica usada foi 0 é falso e 1 é verdadeiro.

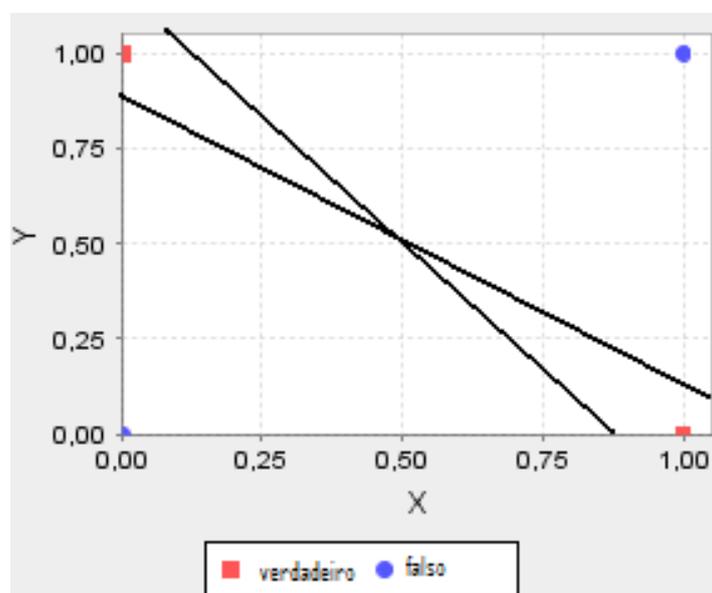


Figura 6. Função lógica XOR resolvida com Backpropagation

2.4.3 Algoritmo Resilient Backpropagation (RPROP)

Para resolver alguns problemas do algoritmo de Backpropagation, surgiu o algoritmo Resilient Backpropagation [4], que se utiliza do gradiente atual e anterior para tomar decisões sobre o

ajuste dos pesos, apenas o sinal do erro é utilizado neste algoritmo, sendo a magnitude do erro substituída por uma regra adaptativa. Na Figura 7 vemos o pseudocódigo do algoritmo Rprop.

O algoritmo se adapta ao gradiente do erro, de forma que se o gradiente anterior tenha o mesmo sinal do gradiente atual é feita uma alteração grande no peso, caso o gradiente anterior tenha sinal contrário ao gradiente atual é feita uma alteração pequena no peso, caso o gradiente atual seja zero a alteração padrão é realizado. Os fatores para grande e pequeno sugeridos pelo autor são respectivamente 0.5 e 1.2, que foram obtidos empiricamente após a execução do RPROP em diversos bases de dados de *benchmark*.

Um das vantagens do RPROP sobre o algoritmo de Backpropagation original é o aprendizado em lote, ou seja, o algoritmo é executado apenas uma vez depois que todos os padrões são ativos na fase de treinamento da MLP.

```

 $\forall i, j : \Delta_{ij}(t) = \Delta_0$ 
 $\forall i, j : \frac{\partial E}{\partial w_{ij}}(t-1) = 0$ 
Repeat
  Compute Gradient  $\frac{\partial E}{\partial w}(t)$ 
  For all weights and biases {
    if  $(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0)$  then {
       $\Delta_{ij}(t) = \text{minimum}(\Delta_{ij}(t-1) * \eta^+, \Delta_{max})$ 
       $\Delta w_{ij}(t) = - \text{sign}(\frac{\partial E}{\partial w_{ij}}(t)) * \Delta_{ij}(t)$ 
       $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$ 
       $\frac{\partial E}{\partial w_{ij}}(t-1) = \frac{\partial E}{\partial w_{ij}}(t)$ 
    }
    else if  $(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0)$  then {
       $\Delta_{ij}(t) = \text{maximum}(\Delta_{ij}(t-1) * \eta^-, \Delta_{min})$ 
       $\frac{\partial E}{\partial w_{ij}}(t-1) = 0$ 
    }
    else if  $(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) = 0)$  then {
       $\Delta w_{ij}(t) = - \text{sign}(\frac{\partial E}{\partial w_{ij}}(t)) * \Delta_{ij}(t)$ 
       $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$ 
       $\frac{\partial E}{\partial w_{ij}}(t-1) = \frac{\partial E}{\partial w_{ij}}(t)$ 
    }
  }
Until (converged)

```

Figura 7. Pseudocódigo do algoritmo Rprop

2.4.4 Algoritmo Cascade Correlation

Um problema ainda não resolvido pelos algoritmos de Backpropagation e Rprop é determinar quantos Perceptrons são necessários para resolve um problema específico. Existe também um problema em utilizar vários Perceptrons por camada, pois ao se ajustar um perceptron os demais são desajustados. Para resolver esses problemas surgiu o algoritmo de Cascade Correlation [7], a principais vantagens são a construção incremental da estrutura da rede e a ausência da fase de retro propagação existente no algoritmo de backpropagation original. Na Figura 8 podemos ver uma rede antes de ser treinada com o CC.

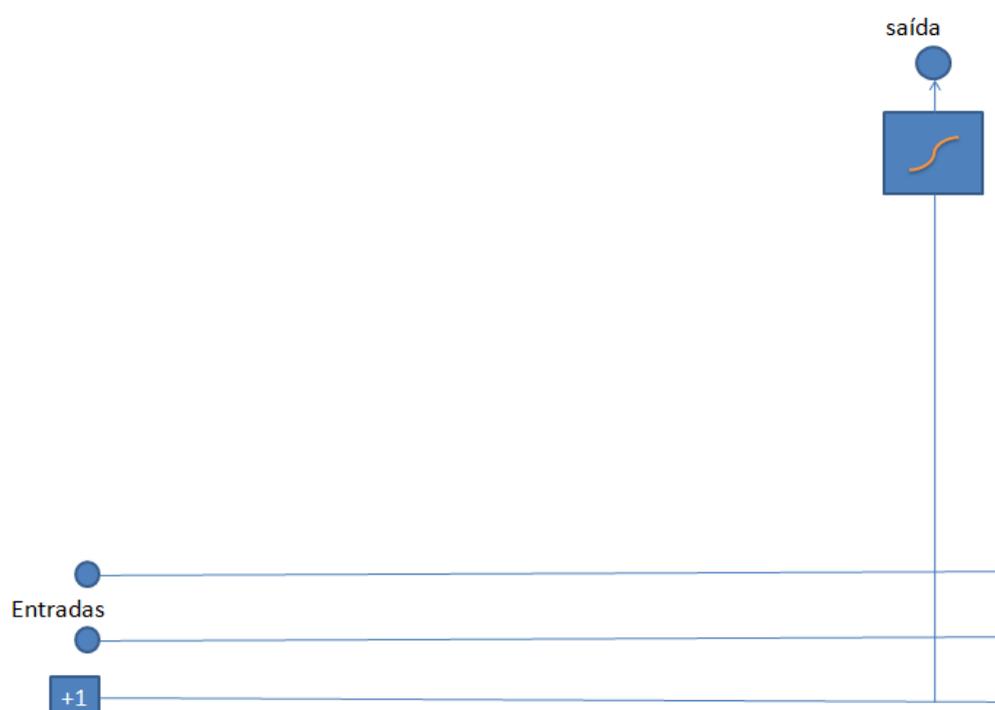


Figura 8. Rede antes do treinamento

A rede sempre começa sem nenhuma unidade intermediária, então a estrutura é treinada e apenas os pesos das conexões da saída são ajustados. Em alguns problemas o algoritmo converge para o erro satisfatório sem adicionar nenhuma unidade intermediária. Caso a estrutura inicial treinada não convergia para o erro satisfatório então são criadas unidades temporárias de diferentes tipos, a fim de explorar vários aspectos da topologia do problema em paralelo, essa unidades no algoritmo recebem o nome de candidatas, e estão conectadas as entradas e as unidades intermediárias já existentes, mas não estão conectadas as unidades de saída. Então os padrões de treinamento são apresentados a rede, as saídas das unidades candidatas são armazenadas para todos os padrões juntamente com os erros de todas saídas, após a execução de todos os padrões é calculada a correlação entre as saídas das unidades candidatas e os erros das unidades de

saída. O treinamento é repetido até que não haja uma variação considerável da correlação das unidades candidatas. Atingida essa condição é escolhida a unidade que possuir a maior correlação em módulo. A melhor unidade então é instalada na estrutura principal da rede de modo que uma cascata de unidades intermediárias é formada, os pesos da nova unidade são fixados no momento da instalação e não serão modificados posteriormente. A instalação das unidades novas em formada de cascata pode ser vista na Figura 9.

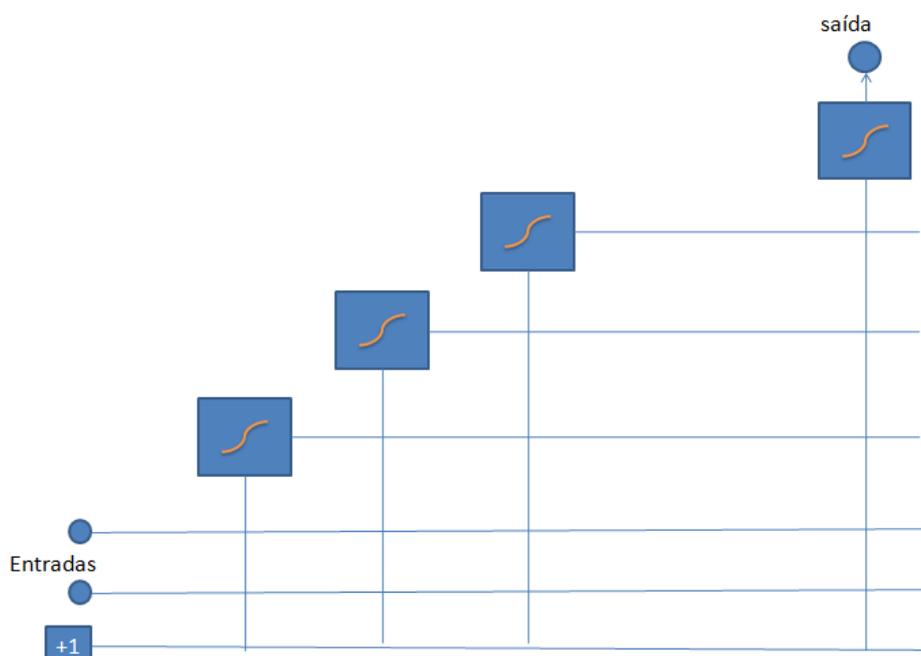


Figura 9. Rede depois do treinamento

A disposição das unidades em forma de cascata, forma filtros de extração de características muito compactos, pois existe apenas uma unidade por camada da rede. Outra vantagem dessa arquitetura é a possibilidade de aprendizado após o primeiro treinamento ser encerrado, por exemplo, uma rede foi treinada para resolver o problema da função lógica XOR, após este é apresentado o problema das duas espirais, veja Figura 10, que é um problema de complexidade muito superior ao XOR. Com a arquitetura em cascata apenas as características ainda não modeladas adicionam novas unidades a cascata de unidades intermediárias. Em alguns problemas semelhantes nenhuma nova unidade é instalada apenas os pesos das saídas são re-configurados, pois todas as características necessárias para resolver o problema semelhante já foram modeladas anteriormente no primeiro treinamento.

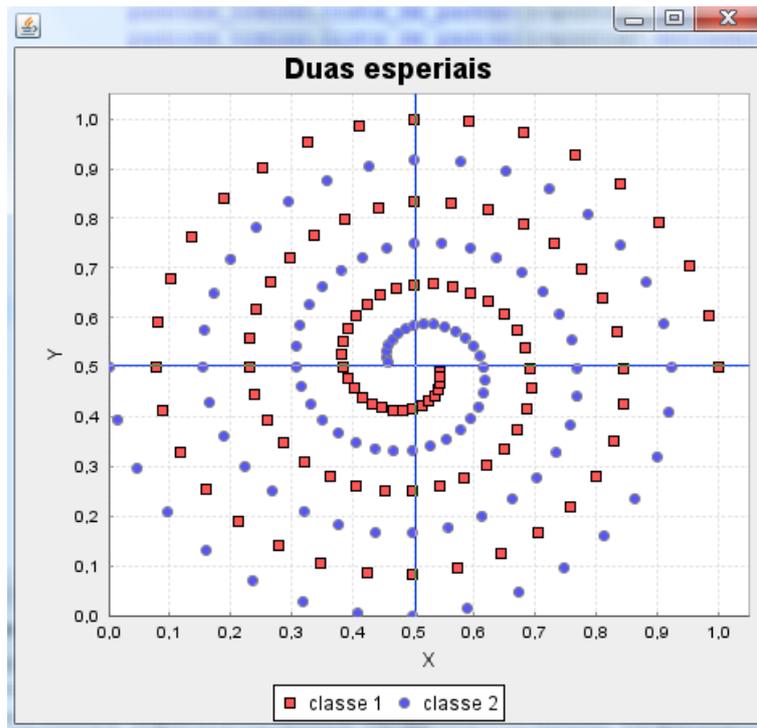


Figura 10. Problemas das duas espirais

O término do treinamento a rede pode ser vista como uma função de mapeamento que modela o problema, sendo possível, por exemplo, testar a rede com valores com os quais a rede não foi treinada, para das duas espirais podemos ver na Figura 11 o resultado para x variando de -2 a 2 e y variando de -2 a 2.

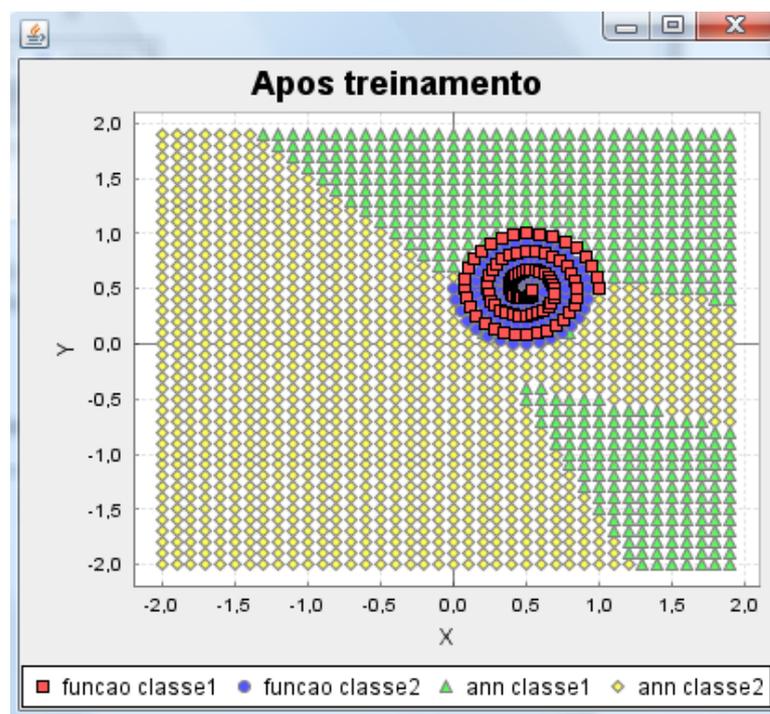


Figura 11. Saída da rede treinada

2.5 Support Vector Machine

É um algoritmo de aprendizagem supervisionada utilizado para classificação e regressão. Considerado um caso especial da regularização de Tikhonov. Uma propriedade especial das SVMs é a capacidade de simultaneamente minimizar o erro empírico de classificação e maximizar a margem geométrica.

Visualizando os dados de entrada como dois conjuntos de vetores em um espaço n-dimensional, a SVM constrói um hiperplano no espaço, veja Figura 13, que maximize a margem entres os dois conjuntos de dados. Para calcular as margens, são construídos dois hiperplanos paralelos, veja Figura 12, um para cada conjunto de dados de forma a marcar o limite do conjunto, então começa o processo para maximizar a distância entre os dois hiperplanos paralelos.

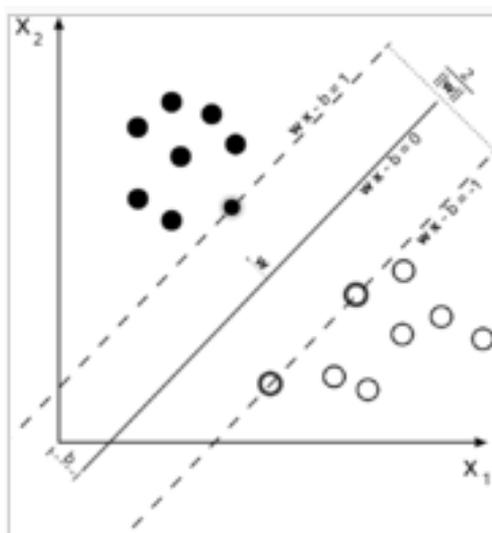


Figura 12. Vetores de dados e hiperplanos paralelos

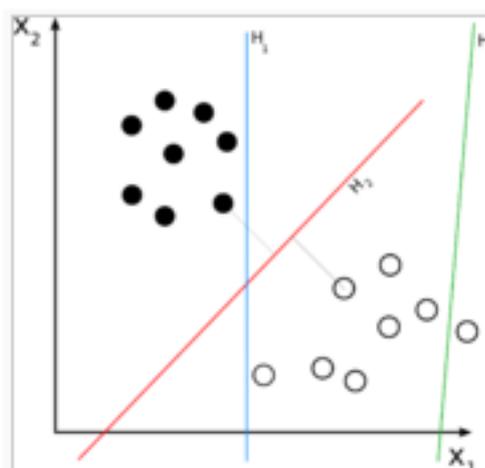


Figura 13. Hiperplano em vermelho maximiza a distância entre vetores dados

Para que o algoritmo de SVM funcione em problemas não lineares é preciso transformar o espaço inicial dos dados para um espaço de maior dimensão, onde os dados iniciais agora são separáveis linearmente. A transformação de espaço é feita por uma função de *kernel* que é aos dados no espaço original e retorna os dados num espaço com mais dimensões. Existem vários tipos de função de *kernel*, sendo polinomial o mais usado para problemas de classificação. Um exemplo de transformação de espaço pode vista na Figura 14, inicial temos duas classes distribuídas sobre um espaço linear, o que torna impossível a separação o único hiperplano, mas após a transformação para um espaço com mais dimensões que o espaço inicial a separação com apenas um hiperplano passa a ser possível.

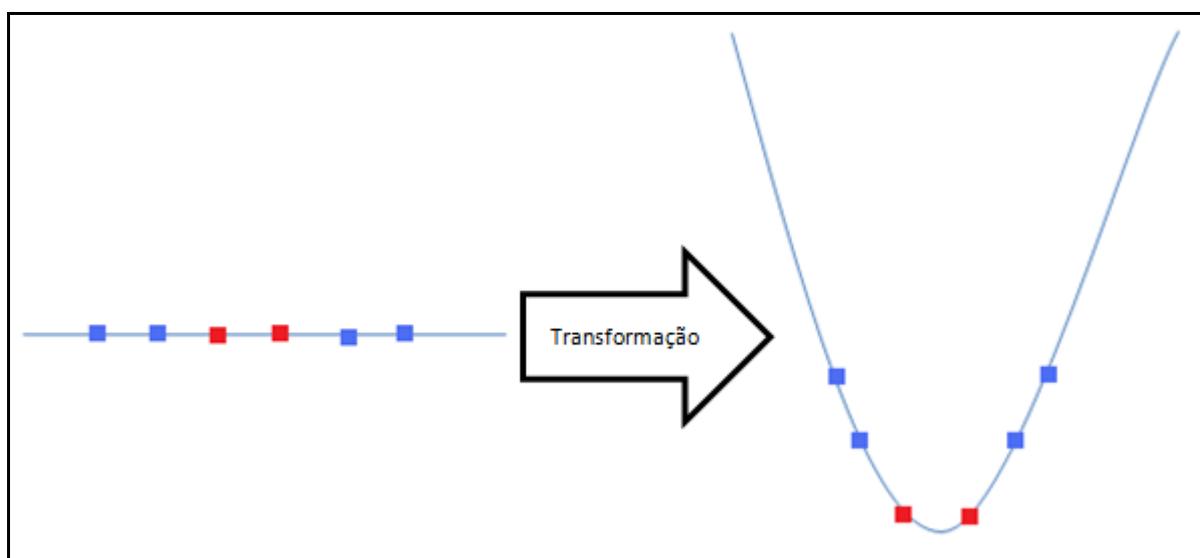


Figura 14. Transformação do espaço usando x ao quadrado

2.6 Agentes Inteligentes

Um agente inteligente tem o desempenho máximo dentro de seu ambiente [14], na Figura 15 temos o esquemático de um agente inteligente. Na Figura 16, temos um esquemático de um agente adaptado para o mercado financeiro.

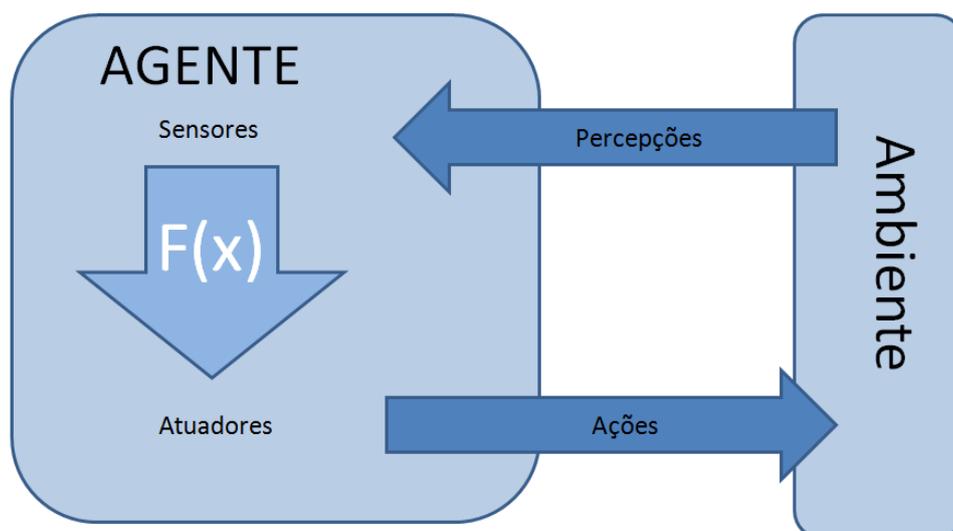


Figura 15. Esquemático de um agente inteligente

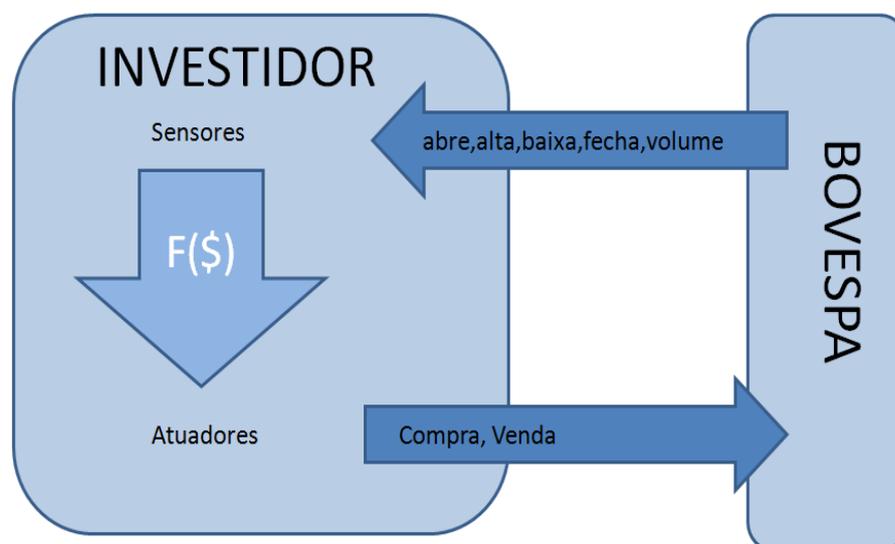


Figura 16. Agente inteligente adaptado para mercado financeiro

2.6.1 Agentes com aprendizado

Agente inteligente que possui capacidade de aprender o comportamento de seu ambiente [14], as melhores técnicas de aprendizado permite ao agente inteligente adaptar ao ambiente, mesmo que o novo ambiente seja totalmente diferente do ambiente inicial em que o agente inteligente foi treinado.

2.6.2 Sistema Multiagente inteligente

É o agrupamento de vários agentes inteligentes autônomos formando uma comunidade inteligente. A sociedade formada por ser colaborativa ou competitiva. Na sociedade colaborativa os agentes cooperem entre si na busca de uma meta comum. Na sociedade competitiva os agentes

competem entre si na busca de uma meta ideal. Sendo o comportamento social dos agentes a base para a inteligência do sistema. O modelo multiagente inteligente é interessante para resolução de problemas complexos onde é preciso conhecimento de vários domínios e que podem envolver dados fisicamente distribuídos.

2.7 Data Mining

É um processo que consiste na aplicação de técnicas para análise de dados e algoritmos de descobertas, onde em tempo aceitável são produzidos padrões ou modelos úteis sobre os dados analisados.

2.8 Análise técnica Financeira

O mercado financeiro possui duas abordagens de análises: fundamentalista e técnica [2]. Este trabalho tem como foco abordagem técnico. A análise técnica financeira é feita através dos indicadores técnicos, que são calculados sobre usando os preços de mercado a fim de gerar gráficos e planilhas que facilitem a análise do mercado financeiro. Existem vários indicadores técnicos [2], o foco de deste trabalho é o indicador *Relative Strength Index*.

2.8.1 Relative Strength Index

O indicador técnico RSI [2] é um dos mais populares osciladores de momento. O RSI compara a magnitude dos ganhos recentes com a magnitude das perdas recentes e retorna a informação como um número entre 0 e 100. O RSI possui apenas um parâmetro, que é o número de períodos de tempo que serão usados para calcular o indicador. O autor do RSI, sugere o uso de períodos de 14 dias. O procedimento para realização do cálculo é exibido na Figura 17.

$$RSI = 100 - (100 / (1 + RS))$$

$$RS = \text{Ganho médio} / \text{Perda média}$$

$$N = \text{tamanho da janela de tempo}$$

$$\text{Ganho médio} = [(\text{Ganho médio anterior} * N - 1) + \text{Ganho médio}] / N$$

$$\text{Primeiro Ganho médio} = \text{Total de ganhos durante últimos } N \text{ períodos} / N$$

$$\text{Perda média} = [(\text{Perda média anterior} * N - 1) + \text{Perda média}] / N$$

$$\text{Primeira Perda média} = \text{Total de perdas durante últimos } N \text{ períodos} / N$$

As perdas devem estar em módulo

Figura 17. Procedimento para cálculo do RSI

Para simplificar o processo de cálculo o RSI é dividido em componentes menores, são eles: RS, ganho médio e perda média. Para calcular os valores do RSI para um conjunto de dados, devemos primeiramente encontrar a magnitude de todos os ganhos e perdas para um dado período N.

É importante entender que o RSI é calculado através do deslocamento de uma janela de tempo de período N através do conjunto de dados. A precisão do cálculo de quantos RSI já foram calculados no passado. O primeiro valor é uma estimativa, os valores subsequentes melhoram esta estimativa. Para um período N é recomendado se calcular pelo menos 2N RSIs para ter uma estimativa apurada. Para iniciar o cálculo, o ganho médio é calculado como o total dos ganhos durante um período N dividido por N. De modo análogo a perda média é calculada como a magnitude total das perdas durante um período N dividido por N. Os próximos valores são calculados utilizando o valor anterior multiplicado por N-1 mais o próximo ganho (ou perda) dividido por N. Esta técnica é conhecida como nivelamento de Wilder.

O RS é simplesmente o valor do ganho médio dividido pela perda média para cada período. O passo final é converter o RS em um oscilador que varia entre zero e 100 usando a fórmula $RSI = 100 - (100 / RS + 1)$. Um exemplo do RSI aplicado a um ativo pode ser visto na Tabela 2, com período de 14 dias.

Tabela 2. RSI aplicado a um ativo usando janela de 14 dias.

dia	fechamento	mudança	ganho	perda	ganho médio	perda média	RS	RSI
0	46,1250							100,0000
1	47,1250	1,0000	1,0000					100,0000
2	46,4375	-0,6875		0,6875				100,0000
3	46,9375	0,5000	0,5000					100,0000
4	44,9375	-2,0000		2,0000				100,0000
5	44,2500	-0,6875		0,6875				100,0000
6	44,6250	0,3750	0,3750					100,0000
7	45,7500	1,1250	1,1250					100,0000
8	47,8125	2,0625	2,0625					100,0000
9	47,5625	-0,2500		0,2500				100,0000
10	47,0000	-0,5625		0,5625				100,0000
11	44,5625	-2,4375		2,4375				100,0000
12	46,3125	1,7500	1,7500					100,0000
13	47,6875	1,3750	1,3750					100,0000
14	46,6875	-1,0000		1,0000	0,5848	0,5446	1,0738	51,7787
15	45,6875	-1,0000		1,0000	0,5430	0,5772	0,9409	48,4771
16	43,0625	-2,6250		2,6250	0,5043	0,7234	0,6970	41,0734
17	43,5625	0,5000	0,5000		0,5040	0,6718	0,7502	42,8634
18	44,8750	1,3125	1,3125		0,5617	0,6238	0,9005	47,3818
19	43,6875	-1,1875		1,1875	0,5216	0,6640	0,7855	43,9921

Quando o ganho médio é maior que a perda média, o RSI sobe por causa que o RS será maior que 1. Analogamente, quando a perda média é maior que o ganho médio, o RSI desce por causa que o RS é menor que 1. A última parte da fórmula do RSI garante que o indicador oscile entre 0 e 100. Note que se a perda média for zero então o RSI será 100 por definição.

Quanto maior a quantidade for o histórico de dados usado para calcular o RSI, mais preciso será o resultado. O fator de nivelamento é calculado continuamente obtendo todos os valores de fechamento de um conjunto de dados. Se o cálculo do RSI iniciar no meio de um conjunto de dados existente, os valores serão apenas aproximados em relação ao RSI completo.

2.8.2 Usando RSI por valor

O criador do RSI sugere que o mercado está comprado quando o valor do RSI está acima de 70 e que o mercado está vendido quando o valor do RSI está abaixo de 30. Se o RSI sobe acima de 30 o mercado está em uma época de oportunidades de compra, se o RSI desce abaixo de 70 o mercado está em uma época de oportunidades de venda. Na Figura 18, podemos ver o RSI calculado com 4 janelas de 4 tempo diferentes, quanto menor a janela de tempo mais sensível o RSI se torna, ou seja, as oscilações entre 0 e 100 serão mais bruscas.

2.8.3 Usando RSI por cruzamento de linha central

A linha central do RSI está no valor 50. Leituras acima ou abaixo podem indicar sinais de compra ou venda. No geral, leituras acima de 50 indicam que a média dos ganhos é maior que a média das perdas, leituras abaixo de 50 indicam que as perdas são maiores que os ganhos.



Figura 18. RSI calculado com janelas de 7,14,21 e 28 dias

Capítulo 3

Metodologia de Estudo

Neste capítulo é apresentada a metodologia que foi aplicada para obtenção dos resultados demonstrados nos estudos de casos.

3.1 Obter dados históricos

Os dados históricos da BOVESPA começaram a ser colocados diariamente a partir 1986 e armazenados em banco de dados, totalizando cerca de 3 milhões de registros.

Na primeira filtragem somente os dados a partir de 1994 e com a moeda Real e mercado a vista serão selecionados, resultando em 853 mil registros de 2700 ativos.

Na segunda filtragem somente os ativos que possuem pelo menos 30 pregões e foram negociados nos último pregão serão selecionadas, resultando em 254 mil registros em 390 ativos.

Na terceira filtragem é realizada a normalização dos preços a fim de contabilizar operações de divisão e agrupamento entre dois pregões seguidos.

3.2 Ordens de compra e venda usando regra D+1

Após a normalização dos preços, é criada uma regra de compra e venda baseada na previsão do dia seguinte, que é feita consulta o histórico no próximo dia. A regra consiste em: Caso preço de amanhã for maior que o preço de hoje então compre hoje, caso contrario venda hoje. Esta regra é utilizada apenas como uma métrica de desempenho, visto que seu uso é inviável na prática.

3.3 Calcular *RSI*

Com os preços normalizados, calcular o *RSI* com períodos de 7, 14, 21 e 28 dias. Os valores gerados serão utilizados pelo sistema multiagente competitivo para encontrar qual a melhor combinação de janela de tempo para o *RSI*, limiar de compra e limiar de venda. O critério para eleição do melhor agente é soma da quantidade de dinheiro em caixa mais a quantidade de dinheiro investido, o agente que tiver o maior total ao fim do último dia de simulação será o vencedor.

3.4 Relação entre *RSI* e ordens de compra e venda usando regra *D+1*

Criar dos padrões de treinamento da rede neural artificial utilizando os valores de *RSI* e a ordem de compra e venda. Formando padrões no seguinte formato:

$$(RSI-7, RSI-14, RSI-21, RSI-28) = \textit{compra ou venda}$$

Após a criação os padrões é realizado o treinamento da rede neural artificial usando alguns dos algoritmos apresentados na fundamentação teórica, como resultado do treinamento é obtida uma função de mapeamento entre os *RSIs* e as operações de compra e venda.

O processo de treinamento é realizado utilizando 75% do total dos padrões, o teste é realizado utilizando os 25% restante que não foram utilizados no processo de treinamento. Para um melhor entendimento do processo de aprendizado veja a Figura 19, que mostra a sequência dos passos tomados para realizar o treinamento.

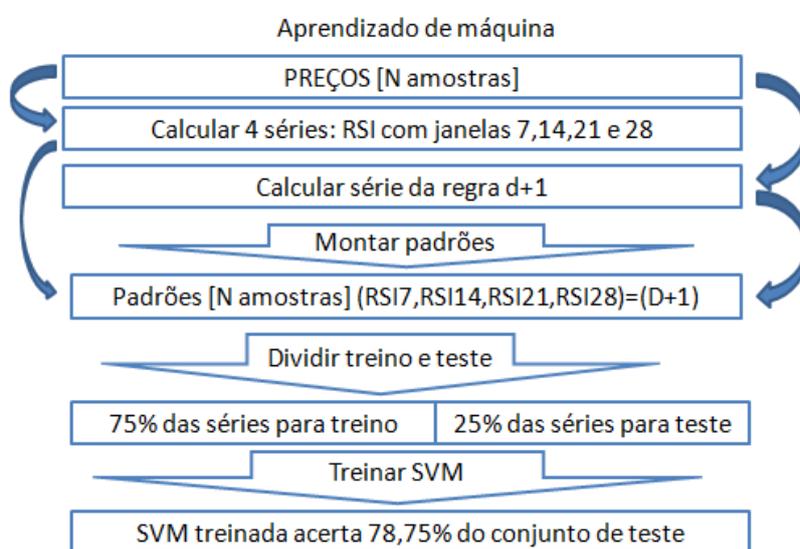
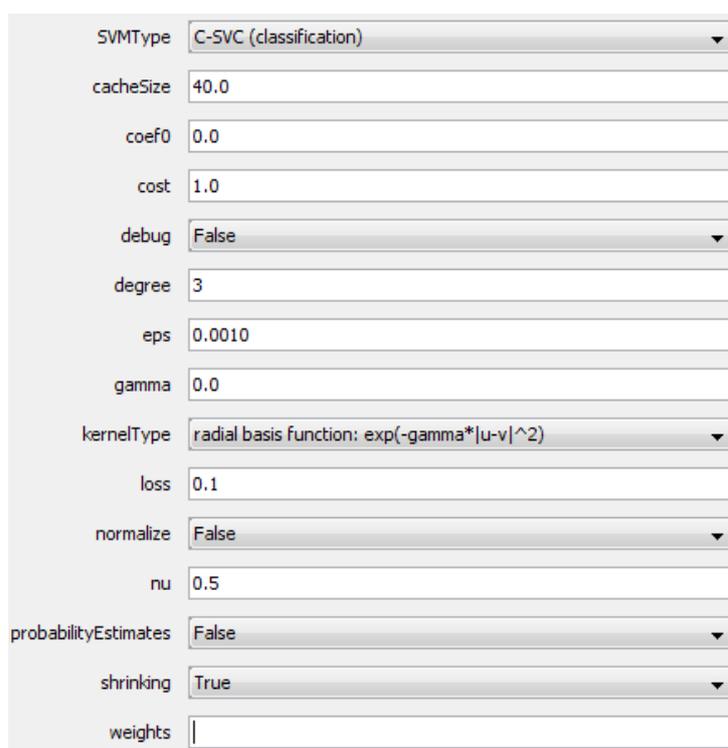


Figura 19. Passos tomados para realizar o treinamento

3.5 Configuração e execução da SVM

Neste trabalho foi utilizado a ferramenta de mineração de dados WEKA em conjunto com a LIBSVM. A configuração utilizada nos experimentos pode ser vista na Figura 20. A LIBSVM possui um recurso [8] que permite encontrar automaticamente os melhores parâmetros para SVM através de validação cruzada, quando realizada a otimização dos parâmetros da SVM provoca um ganho substancial no desempenho de classificação. Devido a massiva quantidade de dados que precisam ser processados em um curto intervalo de tempo, cerca de 400 ativos em menos de 8 horas, o recurso de procura automática de parâmetros não foi utilizado, pois utilizando validação cruzada o tempo de processamento excede 8 horas. O limite de 8 horas é basicamente o tempo entre a liberação do arquivo de transações da BOVESPA do dia anterior e o início do pregão do próximo dia.



SVMType	C-SVC (classification)
cacheSize	40.0
coef0	0.0
cost	1.0
debug	False
degree	3
eps	0.0010
gamma	0.0
kernelType	radial basis function: exp(-gamma* u-v ^2)
loss	0.1
normalize	False
nu	0.5
probabilityEstimates	False
shrinking	True
weights	

Figura 20. Configuração do LIBSVM no WEKA

3.6 Simulação

Para cada ativo é criado um agente inteligente que simula o fluxo de caixa do investidor, formando assim um sistema multiagente competitivo, onde o vendedor é o agente que tiver a maior soma do dinheiro em caixa mais dinheiro investido. Serão feitas 146 simulações por ativo, sendo uma usando regra D+1, uma usando a rede neural artificial e mais 36 combinações de

limiares de compra e venda por RSI, vide Tabela 3. Os resultados das simulações se encontram no apêndice deste trabalho.

Tabela 3. Limiares de compra e venda para RSI

Compra	Venda	Compra	Venda	Compra	Venda	Compra	Venda
10	20	20	40	30	70	50	70
10	30	20	50	30	80	50	80
10	40	20	60	30	90	50	90
10	50	20	70	40	50	60	70
10	60	20	80	40	60	60	80
10	70	20	90	40	70	60	90
10	80	30	40	40	80	70	80
10	90	30	50	40	90	70	90
20	30	30	60	50	60	80	90

Para um melhor entendimento do processo de simulação veja a Figura 21, como é feito o processo de simulação. Esse processo é executado uma vez para cada ação da BOVESPA, são cerca de 340 ações, ou seja, 340 processos de simulação a serem executados.

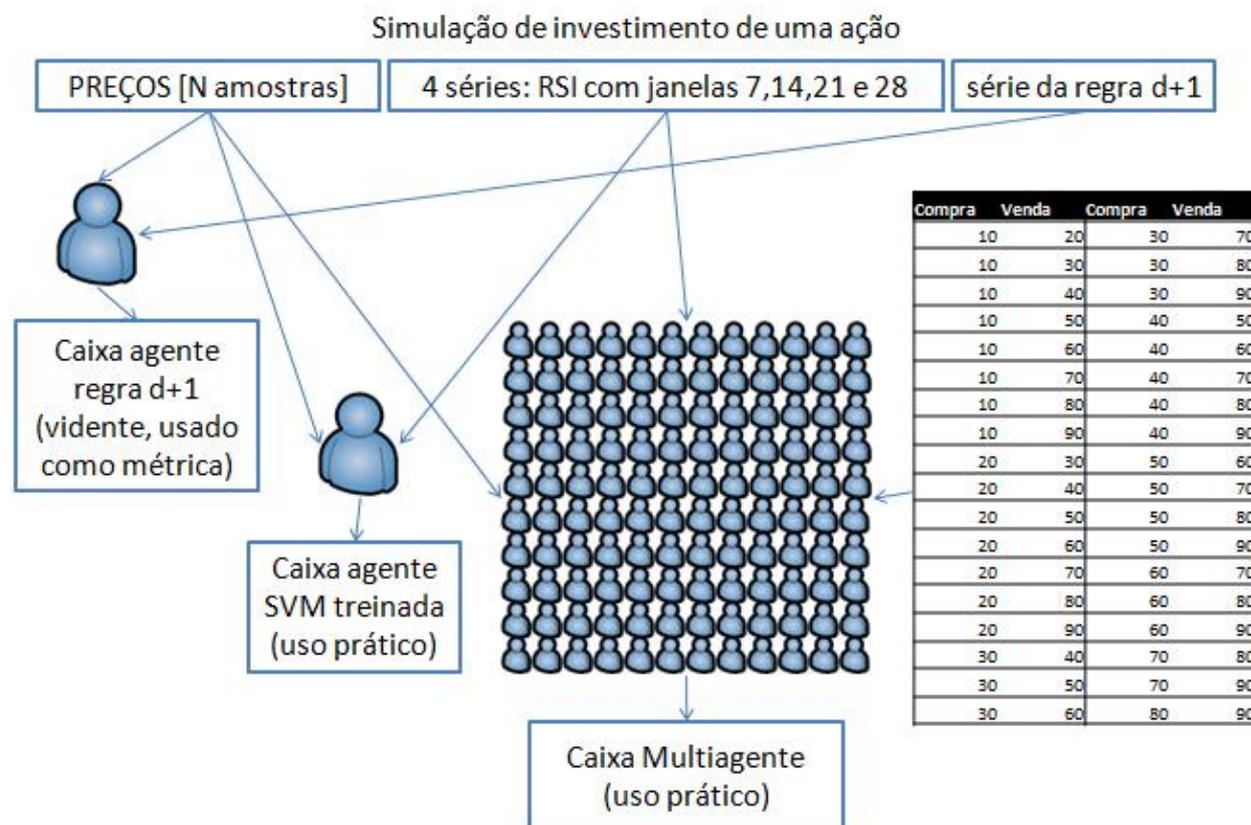


Figura 21. Como é feito o processo de simulação para uma ação

Capítulo 4

Estudo de Caso

Neste capítulo, são discutidos os resultados da metodologia de estudo aplicada ao mercado financeiro. Para o estudo de caso foram selecionados ativos populares na BOVESPA em 2008 para o mercado a vista.

4.1 Preprocessamento dos dados históricos

Inicialmente são gerados gráficos de velas para todos as ações, na Figura 22 temos um exemplo desse tipo de gráfico. Antes de iniciar a análise dos dados é necessário realizar um preprocessamento para verificar se a necessidade de um algum ajuste sobre os preços do ativo. Os problemas ocorrem devido a ausência do fator de divisão ou agrupamento na base de dados disponível pela bolsa de valores. Na Figura 23 vemos o problema que ocorre quando existe supervalorização de um ativo, ou seja, o preço de um único papel é muito alto para ser negociado no mercado, então é feita uma operação financeira pela bolsa de valores que divide um papel em dois um mais sem prejuízo aos seus detentores. Por exemplo, um ativo cuja cotação de papel é R\$ 60,00 é dividido em 3, então o preço de mercado passa a ser R\$20,00 e os investidores que possuem o papel passa a ter um terço da quantidade de papeis do ativo. Na Figura 24 vemos o problema que ocorre quando é feito o agrupamento de papeis de um ativo, por exemplo um ativo de cotado no mercado a R\$ 11,00 é agrupado de forma a juntar 10 papeis em 1, passando a ser cotado a R\$110,00 e os investidores passa a quantidade de papeis reduzida em 10 vezes.

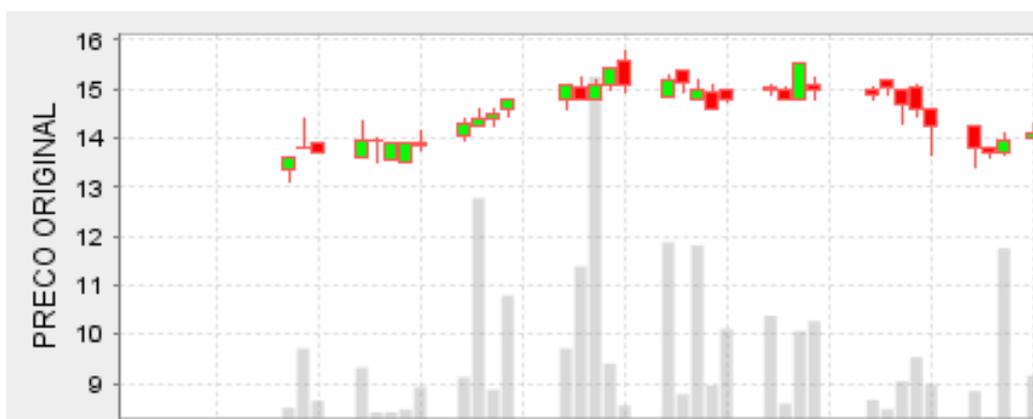


Figura 22. Visualização dos preços de um ativo em gráfico candlestick

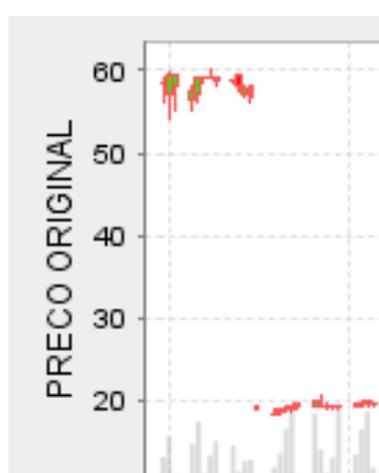


Figura 23. Problema com divisão no preço de um ativo de 1 para 3



Figura 24. Problema com o agrupamento no preço de um ativo de 10 para 1

Outros problemas são seqüência de preços incompleta ou corrompida. Em alguns ativos faltam vários dias seguidos de dados.

4.2 Normalização de preços

Para minimizar os problemas encontrados com os preços na base de dados históricos é realizado um processo de normalização nos preços de forma que o último preço tem seu valor conservado e os valores anteriores são multiplicados por uma constante de proporcionalidade. Na Figura 25 podemos observar que as descontinuidades foram removidas após a normalização. Em alguns casos após a normalização os preços ficaram menores que a menor unidade monetária, mas isso não é um problema para a simulação de caixa que será feita, pois na simulação todas as casas decimais dos preços são usadas. O processo é feito analisando o preço de fechamento do dia passado e o preço de abertura do próximo dia, de forma que a diferença entre esses dois valores deve ser nula ou não ultrapassar um limiar de 10% em módulo do valor de fechamento do dia passado. Caso essa condição não seja atendida a mudança de escala será realizada.

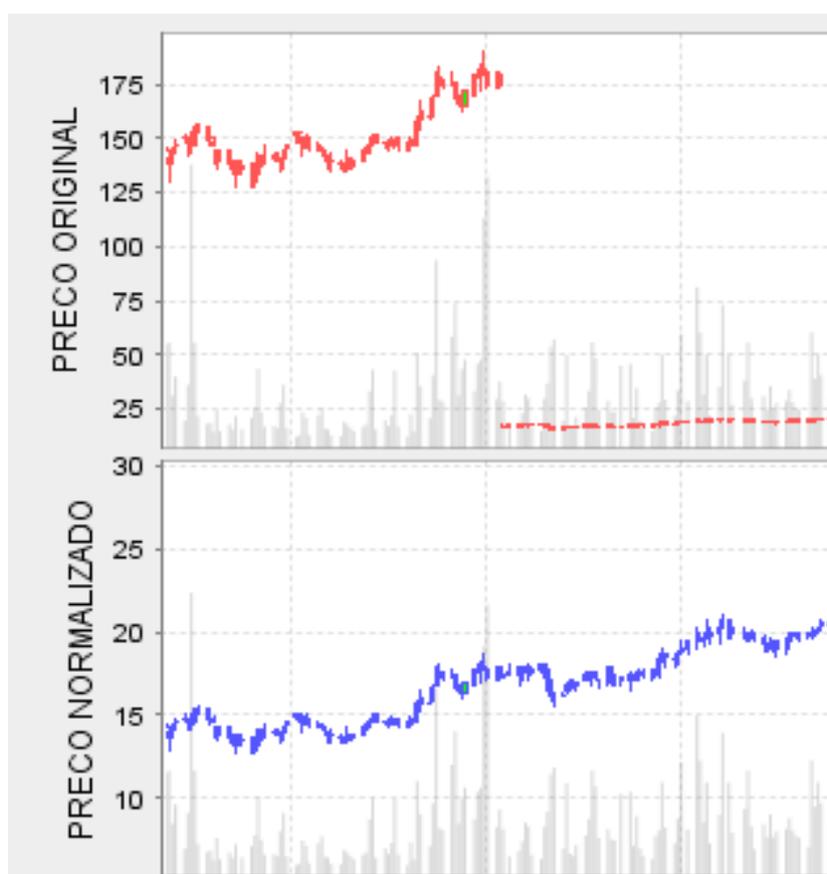


Figura 25. Processo de normalização atribuído ao último valor escala 1.

4.3 Sistema multiagente inteligente por ativo

Cada ativo é simulado por um sistema multiagente inteligente, composto por 146 agentes inteligentes. A composição da sociedade dos agentes inteligentes é composta por um agente que

utiliza a regra D+1, um agente que utilizando uma SVM treinada com vários RSIs e a regra D+1, e 144 agentes usando análise sem simples de RSI com os limiares descritos na Tabela 3.

Os agentes criados para usando SVM, aproximadamente 398 agentes no total sendo 1 por ativo, possuem um média de acerto de 78,75% sobre os padrões de aprendizado. É possível visualizar uma matriz de confusão típica do agente na Figura 26.

```
2008/05/18 17:35:05 Thread-3 Support Vector Machine (SVM)
2008/05/18 17:35:06 Thread-3
=== Confusion Matrix ===
 
 a b c d <-- classified as
206 46 0 0 | a = SOBE
32 230 0 0 | b = DESCE
3 4 0 0 | c = IGUAL
0 1 0 0 | d = IMPREVISIVEL

2008/05/18 17:35:06 Thread-3 percentual correto=83.52490421455938%
```

Figura 26. Matriz de confusão típica do agente inteligente usando SVM.

4.4 Analisando resultados

Após a execução dos sistemas multiagente para todos ativos, precisamos escolher o melhor agente na coleção de agentes baseados em análise simples de RSI para cada sistema multiagente. A escolha será feita considerando que o melhor agente é aquele que possui o maior valor acumulado somando o valor em caixa e o valor investido em ações.

Com esses resultados podemos montar uma tabela com os valores médios de desempenho para cada um dos tipos de agentes contido nos multiagentes. O desempenho é medido pelo percentual de rendimento mensal. O resultado dos experimentos pode ser visto na Tabela 4. Os resultados da MLP e CC usando backprop e rprop são parciais, pois em alguns casos o tempo de treinamento máximo foi excedido, antes que os resultados fossem alcançados.

Tabela 4. Resultado dos experimentos.

Tipo de agente	Acerto da previsão	Rendimento mensal do caixa
Regra d+1	100,00%	60,07%
SVM	78,75%	38,52%
IBOVESPA	-	5,33%
RSI multiagente competitivo	52,21%	3,59%
MLP Backprop	43,98%	~ -2,11%
MLP Rprop	48,45%	~ -1,34%
CC Backprop	72,54%	~ 35,32%
CC Rprop	81,32%	~ 42,11%

Com base nesses resultados podemos construir um agente inteligente que recebe com entrada as ações de todos agentes baseados em SVM e fazer uma interface com um simulador de bolsa de valores, para negociar automaticamente as ações. Na Figura 27 vemos o agente inteligente conectar ao sítio do Investidor Virtual e negociar automaticamente ações. De forma que o lucro médio do investidor tenha desempenho semelhante ao SVM exibido na Tabela 4.

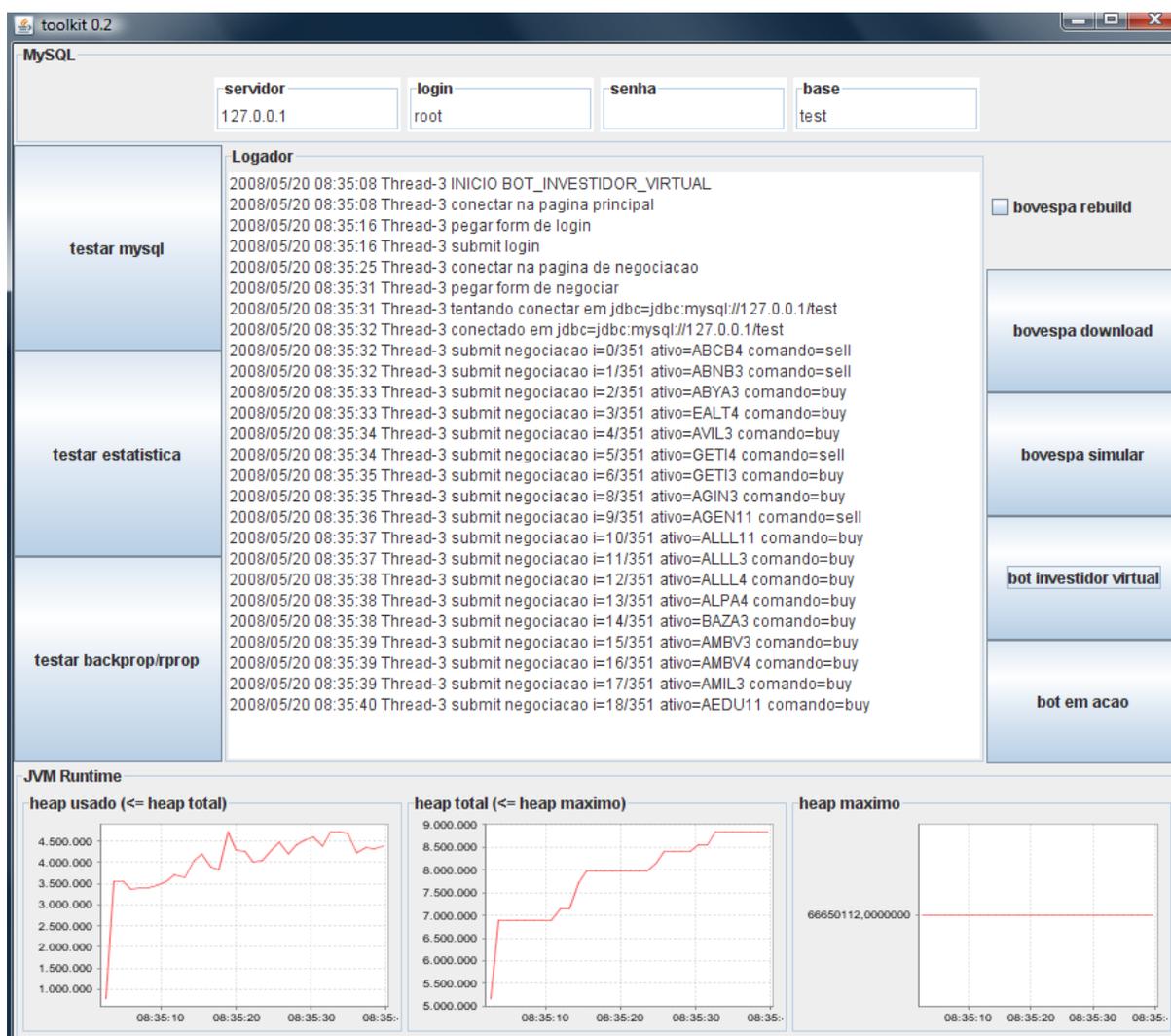


Figura 27. Agente inteligente negociando no Investidor Virtual

Capítulo 5

Conclusões e contribuições

5.1 Conclusão

O sistema proposto neste trabalho atingiu o objetivo principal da sua proposta, que era maximizar os lucros dos investidores, através da utilização de sistemas multiagente inteligentes que utilizam técnicas de redes neurais artificiais para implementar aprendizado de máquina.

Algoritmos com Backpropagation, RPROP, Cascade Correlation foram implementados para serem testados neste problema de previsão, também foram usadas algumas implementações de terceiros como LIBSVM e WEKA.

O artefato final produzido um agente inteligente que combinando todos os subsistemas de forma que o progresso de negociação de compra e venda de ações com lucro maximizado foi totalmente automatizado. Os resultados experimentais indicam que o agente inteligente acerta em 78,75% de suas previsões.

Este trabalho é relevante para a área financeira, pois gerou sistemas automáticos capazes de maximizar os lucros dos investidores mais experientes e minimizar o prejuízo dos investidores iniciantes.

Uma rede neural artificial treinada pode ser vista como uma função que recebe dados do passado e retorna a operação ótima para o presente, de forma a maximizar o desempenho da solução no problema abordado. Também pode ser vista como um interpelador universal adaptativo, sendo capaz de aprender qualquer coisa para que fosse treinada.

5.2 Problemas encontrados

Obtenção de dados confiáveis e consistentes do sistema financeiro é uma das maiores dificuldades encontradas. Pois os dados disponíveis pela própria BOVESPA possuem problemas de normalização, preços faltando, campos corrompidos entre tantos outros.

As redes neurais com estrutura fixa se mostraram ineficientes no problema abordado neste trabalho após testes com redes com uma e duas camadas usando até 30 perceptrons em cada camada, então foi testado o algoritmo construtivo cascade correlation, que resolveu o problema com média de 93 perceptrons em cascata em ações com menos de 1000 padrões, mas o tempo de treinamento para ações com mais de 1000 padrões passou a exceder o tempo máximo de treinamento antes que os resultados fossem obtidos. Portanto foi usado SVM para que o tempo de treinamento fosse inferior a 24horas, para permitir o sistema negociar diariamente no simulador de bolsa de valores.

5.3 Contribuições

Modificações nos algoritmos de treinamento Backpropagation [3] e Resilient Backpropagation [4] que evitam mínimos locais, ou seja, encontrar o mínimo global das funções de erro, utilizando algumas idéias dos algoritmos Cascade Correlation [7], Support Vector Machine [8] e Particle Swarm Optimization [9].

Um sistema para investimento automático em simuladores de bolsa de valores que maximiza o retorno do investidor.

5.4 Trabalhos Futuros

Em trabalhos futuros é proposto a utilização de outros indicadores técnicos além do RSI, incluir cálculos de corretagem ao sistema de simulação, estender o suporte para todas bolsa de valores do mundo, e um sistema de priorização de carteira que aloque recursos financeiros automaticamente de forma a maximizar o lucro mesmo quando os recursos financeiros são limitados.

Bibliografia

- [1] NANNI, Loris. **Multi-resolution Subspace for Financial Trading**. Pattern Recognition Letters, v. 27, pp. 109-115, 2006.
- [2] WILDER, J. Welles, **New Concepts in Technical Trading Systems**. 1 Ed. Wiley, 1978
- [3] ROJAS, Raúl, **Neural Networks - A Systematic Introduction**. Springer-Verlag, Berlin, New-York, 1996
- [4] RIEDMILLER, Martin. **RPROP - A Fast Adaptive Learning Algorithm**. International Symposium on Computer and Information, 1992, pp. 586-591
- [5] BOVESPA, **Layout do arquivo da Série Histórica de Cotações**. Disponível em: http://www.bovespa.com.br/Pdf/SeriesHistoricas_Layout.pdf. Acesso em 20 de março de 2008.
- [6] BARKER, Philip. **Java Methods for Financial Engineering**. 1 Ed, Springer, 2007.
- [7] FAHLMAN, Scott E. **The Cascade-Correlation Learning Architecture**. Advances in Neural Information Processing Systems, 1990. pp. 524-532
- [8] CHANG, Chih-Chung. **LIBSVM: a library for support vector machines**, Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. Acesso em 20 de março de 2008
- [9] JIANBO, Yu et al. **An Improved Particle Swarn Optimization for Evolving Feedforward Artificial Neural Networks**, Neural Process Letters, 26(3), pp. 217-231, 2007
- [10] CYBENKO, G.V. **Approximation by Superpositions of a Sigmoidal function**, Mathematics of Control, Signals and Systems, vol. 2 no. 4 pp. 303-314, 1989
- [11] Siegelmann, H. T. and Sontag, **On the computational power of neural nets**. J. Comput. Syst. Sci. 50, 1 (Feb. 1995), 132-150. E. D. 1995.
- [12] Hebb, D.O., **The organization of behavior**, New York: Wiley, 1949
- [13] Scott E. Fahlman. **Faster-learning variations on back-propagation: An empirical study**. In T. J. Sejnowski G. E. Hinton and D. S. Touretzky, editors, 1988 Connectionist Models Summer School, San Mateo, CA, 1988. Morgan Kaufmann.
- [14] Russell S, Norvig P. **Artificial Intelligence: A Modern Approach**, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey, 1995

[15] STOCKCHARTS, **Introduction to Technical Indicators and Oscillators**, Disponível em:
http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:introduction_to_tech#oscillator_types. Acesso em 28 de maio de 2008.

[16] INFOMONEY, **Análise técnica: conceitos básicos**, Disponível em:
<http://web.infomoney.com.br/templates/news/view.asp?codigo=%20360077&path=/investimentos/acoes/analise/tecnico/>. Acesso em 28 de maio de 2008.

[17] BRASIL, Banco Central. **Publicações**, Disponível em:
<http://www.bcb.gov.br/?PUBLICACOES>. Acesso em 28 de maio de 2008.

Apêndice A

Neste seção estão incluídos alguns materiais produzidos para este trabalho

