

ANGELO LUIS FRANGOULIDIS TRINDADE
BRUNNO VODOLA MARTINS
RODRIGO LACERDA DE OLIVEIRA

FEPREM
ferramenta de previsão de fundos multimercado

São Paulo
2013

ANGELO LUIS FRANGOULIDIS TRINDADE
BRUNNO VODOLA MARTINS
RODRIGO LACERDA DE OLIVEIRA

FEPREM
ferramenta de previsão de fundos multimercado

Trabalho de Conclusão de Curso apresentado à Escola
Politécnica da Universidade de São Paulo

Área de Concentração: Sistemas Eletrônicos
Orientador: Prof. Emilio Del Moral Hernandez

São Paulo
2013

RESUMO

Nosso produto, FEPREM, possibilita que gestores de patrimônio possam acompanhar detalhadamente como estão alocados os recursos investidos em fundos multimercado entre dez classes de ativos, além de poder acompanhar as rentabilidades de fundos de investimento do Brasil. Tudo isso em uma plataforma Web. Para avaliação da precisão dos dados de saída referentes à exposição por classe de ativo, há medidas estatísticas de erro. A nossa previsão de exposições por classe de ativo será solucionada a partir de uma rede neural, uma vez que estamos trabalhando com relações altamente não lineares entre as variáveis disponíveis. O gestor de patrimônio, nosso cliente, não terá de se preocupar em aprimorar nossas informações, nem de ter conhecimentos sobre redes neurais, forneceremos a solução completa. Quanto à comercialização, nosso produto será de fácil distribuição, pois não haverá necessidade de instalar ou implementar nada fisicamente no computador do usuário. Todo acesso será via internet, assim como o controle de quem pode realizar o acesso. É importante também destacar que nosso cliente não terá de se preocupar com a confidencialidade da identidade de seus investidores, uma vez que no nosso sistema não constará nenhum nome de cliente ou família.

Palavras-chave: Engenharia, Engenharia eletrônica, Redes neurais, Gestão de patrimônio, Fundos multimercado.

ABSTRACT

Our product, FEPREM, makes it possible for asset managers to follow up the invested resources allocation from hedge funds between eleven classes of assets, besides being able to visualize profitability's from major Brazilian hedge funds. All wrapped in a Web platform. For the output data precision assessment, there will be a set of statistic error measurement graphics. The forecast of exposures divided by asset class will be resolved with the aid of a neural network, given the highly non-linear relationship between the considered variables. The asset manager, our client, won't have to worry about enhancing the information presented in our platform, nor will he have to have any kind of knowledge concerning neural networks. We will supply him with a complete solution. About the commercialization, our product will be easily distributed, as it will not be necessary to install or implement any kind of device physically in the client's computer. The entire access will be granted by internet, so as the authorization system. It is also important to highlight that the client will not have to worry about confidentiality when it comes to the identity of his investors, given that there will be no client name stored in our database.

Keywords: Engineering, Electronics engineering, Neural networks, Asset management, Hedge funds, Wealth management

LISTA DE ILUSTRAÇÕES

Figura 1 - Concentração por classe de ativo	13
Figura 2 - Concentração por classe de ativo	15
Figura 3 - Estrutura de uma rede neural de três camadas.....	19
Figura 4 - Exemplo de um caso onde uma regressão polinomial com polinômio de grau muito alto prejudica o resultado.....	21
Figura 5 - Detalhamento das classes de ativos de renda fixo	22
Figura 6 - Rentabilidade de classes de ativos de renda fixa em 2012	24
Figura 7 - Rentabilidade de classes de ativos de renda variável em 2012.....	25
Figura 8 - Gráfico detalhando a não linearidade da rentabilidade do mercado de opções.....	28
Figura 9 - Esquema de ciclo de requisição da arquitetura MVC.....	31
Figura 10 - Fluxo do Projeto - FEPREM.....	40
Figura 11 - Fluxo operacional do produto final	41
Figura 12 – Rentabilidade do fundo Simula em relação ao CDI.....	46
Figura 13 - Rentabilidade do fundo Simula x Fundo FIDES Hedge Plus	47
Figura 14 - Concentração do fundo Simula em 06/01/2010	48
Figura 15 - Modelo 1 da rede neural - Fundo fictício.....	50
Figura 16- Entradas e saídas do Modelo 1 de Rede Neural - Valores absolutos	51
Figura 17- Entradas e saídas do Modelo 1 de Rede Neural - Valores nomalizados	51
Figura 18- Rede neural A.....	54

Figura 19- Rede Neural B	55
Figura 20- Rede neural C	56
Figura 21- Rede neural D	56
Figura 22- Tela 1 de inputs.....	58
Figura 23- Tela 2 de inputs.....	59
Figura 24- Fluxo no Matlab.....	60
Figura 25 - Output dos resultados da Rede Neural	62
Figura 26 - Esquema da Camada Model.....	66
Figura 27 - Elementos básicos do framework Bootstrap	68
Figura 28- Exemplo de gráfico de linha gerado pela ferramente Google Charts.....	70
Figura 29- Exemplo de gráfico de barras gerado pela ferramente Google Charts ...	70
Figura 30- Concentração do Fundo Fictício	76
Figura 31 - Concentração Prevista do Fundo Fictício	77
Figura 32- Concetração de dólar no fundo Simula, valores reais e previstos	80
Figura 33- Erro na previsão de concentração de dólar	80
Figura 34- Concretarção do IMA-B5+ no Fundo Simula, valores reais e previstos ...	82
Figura 35- Erro na previsão de concentração do IMA-B5+	82
Figura 36- Concetração do IMA-B5 no Fundo Simula, valores reais e previstos	84
Figura 37 - Erro na previsão de concentração do IMA-B5	84
Figura 38 - Concentração do Índice Indústria no Fundo Simula, valores reais e previstos	86

Figura 39 - Erro na previsão de concentração do Índice de Industria	86
Figura 40- Previsão pelo modelo neural do fundo NEO MULTI ESTRATÉGIA FIM para a classe Índice de Consumo	89
Figura 41- Previsão pelo modelo linear do fundo NEO MULTI ESTRATÉGIA FIM para a classe Índice de Consumo	89
Figura 42 - Previsão pelo modelo neural do fundo NEO MULTI ESTRATÉGIA FIM para a classe Dólar.....	91
Figura 43 - Previsão pelo modelo linear do fundo NEO MULTI ESTRATÉGIA FIM para a classe Dólar.....	91
Figura 44 - Previsão para o fundo NEO MULTI ESTRATEGIA FIM para a classe IMA-B5	92
Figura 45- Previsão pelo modelo neural do fundo ITAU EQUITY HEDGE ADVANCED FIM para a classe Índice de Materiais Básicos.....	93
Figura 46- Previsão pelo modelo linear do fundo ITAU EQUITY HEDGE ADVANCED FIM para a classe Índice de Materiais Básicos.....	94
Figura 47 - Previsão pelo modelo linear do fundo BRZ LONG SHORT FIM para a classe Índice de Energia Elétrica.....	95
Figura 48 - Previsão pelo modelo linear do fundo BRZ LONG SHORT FIM para a classe Índice de Energia Elétrica.....	96
Figura 49 - BRZ LOND AND SHORT - Erro final de previsão	98
Figura 50 - BRZ LOND AND SHORT - Erro final de previsão com modelo linear ...	102
Figura 51 - Tela de variação dos valores das cotas	107
Figura 52 - Tela de exposições de um fundo	108
Figura 53 - Tela de evolução das exposições	110

LISTA DE TABELAS

Tabela 1 - Erros quadráticos médios dos fundo por classe - Redes Neurais	97
Tabela 2- Percentual de fundos por ordem de grandeza de erro - Rede Neural.....	99
Tabela 3 - Erro quadrático médio das redes neurais.....	100
Tabela 4 - Percentual de redes por ordem de grandeza de erro.....	100
Tabela 5 - Erros quadráticos médios dos fundo por classe - Modelamento Linear.	101
Tabela 6 - Percentual de fundos por ordem de grandeza de erro - Modelo Linear .	103
Tabela 7 - Erro quadrático médio - modelamento linear	103
Tabela 8 - Percentual de redes por ordem de grandeza de erro – Modelo Linear..	104
Tabela 9 - Superioridade da modelagem linear.....	105

SUMÁRIO

INTRODUÇÃO	13
1 OBJETIVOS.....	16
2 BACKGROUND	17
2.1 Modelagem Matemática.....	17
2.2 Redes Neurais	18
2.3 Redes Neurais – Saídas	21
2.3.1 Renda fixa: IMA	22
2.3.2 Renda variável.....	24
2.4 Detalhamento do Problema.....	27
2.4.1 Alavancagem.....	29
2.5 Definições gerais sobre a estrutura da plataforma.....	29
2.5.1 A camada <i>Model</i>.....	30
2.5.2 A camada <i>View</i>.....	30
2.5.3 A camada <i>Controller</i>.....	31
2.5.4 Ciclo de requisição da arquitetura MVC	31
2.5.5 Alternativas de linguagens de programação.....	32
2.5.5.1 PHP.....	32
2.5.5.2 C++	33
2.5.5.3 Matlab	33
2.5.5.4 VBA.....	33
2.5.5.5 Python.....	34
2.5.5.6 Alternativa escolhida	34
2.5.6 Alternativas para armazenamento de dados	34
2.5.6.1 PostgreSQL.....	35
2.5.6.2 SQLite	35
2.5.6.3 MySQL	35
2.5.6.4 Alternativa escolhida	36
2.5.7 Interface Gráfica	36
2.5.7.1 HTML	36

	10
2.5.7.2 CSS.....	36
3 ANÁLISE DE VIABILIDADE.....	38
3.1 Modelagem Matemática.....	38
3.2 Viabilidade de Software.....	39
3.3 Viabilidade Econômica	39
3.4 Diagramas de projeto e produto.....	39
4 CRIAÇÃO DO FUNDO FICTÍCIO	43
4.1 Motivação.....	43
4.2 Algoritmo de criação.....	43
4.3 Comparação com fundos reais	45
4.4 Construção de modelos	48
4.4.1 Modelo 1 de rede neural – Fundo fictício.....	49
4.4.2 Modelo 2 de rede neural – Fundo Fictício	52
4.5 Conclusões sobre o fundo fictício	52
5 REDES NEURAIS TREINADAS PARA FUNDOS REAIS.....	54
5.1 Modelos de rede neural utilizados.....	54
5.2 Ferramentas de auxilio de escolha de rede neural	57
5.3 Algoritmos para automatização do processo de treinamento.....	57
5.4 Medidas de erro	61
5.5 Validação	61
5.5.1 Método linear.....	63
6 CRIAÇÃO DO SITE E BANCO DE DADOS	64
6.1 Domínio e servidor de hospedagem.....	64
6.1.1 Framework utilizado.....	65
6.1.2 Detalhes da camada Model	65
6.1.2.1 Model: Fundo	65
6.1.2.2 Model: Cota.....	66
6.1.2.3 Model: Distribuição.....	66
6.1.2.4 Relação entre os modelos.....	66

6.1.3	Ferramentas de interface	67
6.1.4	Interface básica – Bootstrap	67
6.1.4.1	Ferramenta de exibição de gráficos - Google Charts	68
6.2	Aspectos operacionais	70
7	RESULTADOS	73
7.1	Estratégia de testes	73
7.2	Fundo fictício	74
7.2.1	Ensaio 1	75
7.2.1.1	Resultados do Ensaio 1	75
7.2.2	Ensaio 2	77
7.2.2.1	Resultados do Ensaio 2	78
7.2.3	Ensaio 3	80
7.2.3.1	Resultados do Ensaio 3	81
7.2.4	Ensaio 4	82
7.2.4.1	Resultados do Ensaio 4	83
7.2.5	Ensaio 5	84
7.2.5.1	Resultados do Ensaio 5	85
7.2.6	Conclusões sobre os ensaios	87
7.3	Fundos reais	88
7.3.1	Exposição variando num padrão complexo	88
7.3.2	Exposição baixa e aproximadamente linear	90
7.3.3	Variação imprevisível	92
7.3.4	Modelo linear superior ao modelo neural	93
7.3.5	Fundos Long and Short	95
7.4	Análise quantitativa consolidada	97
7.4.1	Rede neural	97
7.4.2	Modelagem linear	101
7.4.3	Comparação entre os dois modelos	104
8	INTERFACE COM O CLIENTE – SITE	106
8.1	Tela de variação dos valores das cotas	106
8.2	Tela de exposições	107
8.3	Tela de evolução de exposição por classe de ativo	109

8.4 Tela de cálculo de rentabilidade da carteira.....	111
9 CONCLUSÃO.....	112
REFERÊNCIAS.....	113

INTRODUÇÃO

Segundo a ANBIMA, no inicio de 2013 havia um volume financeiro de mais de R\$ 527 bilhões pertencente apenas às famílias e indivíduos mais ricos do Brasil, valor que cresceu mais de 20% no decorrer de 2012.

Esses investidores aplicam seus recursos com diferentes objetivos como: aumento de capital, preservação de patrimônio contra a desvalorização monetária ou até mesmo necessidade de rendimentos das aplicações para subsistência. O setor do mercado financeiro que atende esse tipo de cliente é conhecido como *wealth management*.

Tanto bancos como *family offices* atendem esse público, sendo que uma das vantagens das *family offices* é a possibilidade de venda de produtos de diversos bancos, enquanto os bancos só oferecem aos seus clientes os próprios produtos.

Nosso trabalho foi desenvolver um produto (*software online*) designado para suprir uma necessidade dos gestores de patrimônio de *private banks* e *family offices*, a abertura da carteira de fundos multimercado.

Antes de detalhar as características de nosso produto, Apresentamos de forma sucinta na tabela abaixo em quais tipos de ativos ou estratégias (ou gestores de patrimônio) alocam os recursos de seus clientes:

Figura 1 - Concentração por classe de ativo

Tipo de Ativo	Percentual médio alocado pelos gestores (Brasil)
FUNDOS	46,40%
TÍTULOS E VALORES MOBILIÁRIOS	47,30%
CAIXA / POUPANÇA	0,80%
PREVIDÊNCIA ABERTA	5,30%
OUTROS INVESTIMENTOS	0,20%

Fonte: ANBIMA

Nosso produto trata da parte de Fundos, mais especificamente de Fundos de Investimento Multimercado abreviado por FIM¹ e como podemos ver na tabela acima, grande parte dos recursos investidos estão alocados nesse tipo de ativo.

O estrategista (ou gestor de patrimônio) deseja de saber exatamente como estão alocados os recursos de seus clientes, mais especificamente em que tipo de título e em que setor da economia no caso de ações estão os recursos de seus clientes. Quando o gestor vai levantar essas informações, ele se depara com um empecilho: não há como saber em que ativos o gestor do fundo multimercado (FIM) está investindo atualmente, pois essa informação é divulgada com seis meses de *delay*. Além disso é da natureza desses fundos realizar qualquer tipo de investimento, seja em ações de qualquer setor, seja em títulos públicos ou privados.

Colocando de outra forma, para os investimentos diretos (todos investimentos excluindo-se aplicação em fundos) que o estrategista faz, ou seja, as ações e os títulos que ele investe, ele sabe exatamente o quanto de dinheiro está em cada aplicação. No caso de fundos multimercado que ele investe, ele não sabe como estão divididos as aplicações dentro daquele fundo².

Nosso trabalho foi desenvolver uma ferramenta para que os estrategistas saibam como os ativos dos fundos multimercado estão divididos entre dez classes de ativos que serão detalhadas posteriormente. Para tanto utilizamos uma rede neural para modelar a estratégia de cada gestor de fundo multimercado. Além disso, temos uma plataforma interna em que é possível tratar estatisticamente os dados de entrada e saída da rede neural, assim como fornecer medidas de erro. No software é também possível que o usuário consulte as rentabilidades passadas de diversos fundos de investimento.

Com esse produto, o estrategista poderá saber como estão posicionados os fundos em que ele investe e dessa forma realizar mais investimentos no fundo (se acreditar que o posicionamento daquele gestor é promissor) ou desinvestir o

¹ Segundo a ANBIMA, Fundos que realizam operações em diversas classes de ativos (renda fixa, renda variável, câmbio, etc.), definindo as estratégias de investimento baseadas em cenários macroeconômicos de médio e longo prazos, atuando de forma direcional.

² A divulgação da carteira de fundos multimercado é divulgada com *delay* de 6 meses, exceto no caso de haver um contrato entre a gestora do estrategista e o gestor do fundo multimercado.

dinheiro de seu cliente do fundo (caso o estrategista tenha uma visão oposta do gestor do fundo multimercado). Abaixo o gráfico que o estrategista possuirá através de nosso *software* em conjunto com as informações que ele tem de investimentos em ativos que não fundos.

Figura 2 - Concentração por classe de ativo



Fonte: Autor

1 OBJETIVOS

O projeto possui como finalidade mais do que apenas uma modelagem matemática de um Fundo de Investimento Multimercado. Desenvolvemos um produto que consiste em uma ferramenta online para gerenciamento de investimentos, provendo informações e análises matemáticas relevantes, com auxílio de um fundo fictício (criado por nós), para fazer os primeiros ensaios e encontrar modelos parciais. Temos as seguintes funcionalidades:

- Acesso online dos resultados pelo usuário final mediante *login*. Determinadas partes são desenvolvidas em um ambiente local (com Matlab), e os resultados são carregados para o ambiente online.
- Exposição por classe de ativo dentro de um Fundo de Investimento Multimercado.
- Histórico dos erros gerados nas últimas modelagens para o fundo em questão.
- Rentabilidade das classes de ativos usadas para decompor o FIM, apresentadas graficamente.
- Rentabilidades de fundos de investimento do Brasil, apresentadas graficamente.
- Rentabilidade da carteira total de investimento do cliente, apresentada graficamente.

2 BACKGROUND

2.1 Modelagem Matemática

A modelagem matemática de aspectos do mercado financeiro e da economia em geral é um campo amplamente estudado, e proporciona um desafio muito grande por se tratar de fenômenos cujo comportamento é altamente complexo e pouco conhecido. No caso deste trabalho, onde desejamos conhecer em que proporções estão distribuídos os recursos de um fundo multimercado temos a característica de um problema com muitas não linearidades, com variância no tempo e, como citado acima, com poucas informações sobre a natureza do fenômeno.

Para tratar este tipo de problema, precisamos de uma estratégia de modelagem onde se assume muito pouco a priori, ou seja, não informarmos quase nada ao modelo sobre a natureza do problema, e que mesmo assim seja capaz de tratar as não linearidades e a complexidade em questão.

As redes neurais foram escolhidas exatamente por suprir as necessidades citadas acima, no entanto, algumas desvantagens em relação aos modelos mais simples surgem com essa escolha:

- É necessária uma grande quantidade de dados para chegar a bons resultados.
- Deve-se tomar cuidado para utilizar como entrada somente o que é relevante para o problema, e isso quase sempre não é totalmente conhecido.
- Também é importante que se coloque os dados de entrada numa forma conveniente para a rede neural, utilizando técnicas de pré-processamento (com normalização, médias móveis e outras).

A rede neural deve ser construída num tamanho adequado para o problema (número de nós e camadas), pois redes muito grandes necessitam de uma quantidade de dados igualmente grandes, dados que geralmente não estão disponíveis em quantidade arbitrária. Por outro lado, redes muito pequenas podem não modelar o problema adequadamente.

Como consequência da dificuldade que existe para se construir um modelo adequado com redes neurais, é necessário que se tenha um ambiente computacional que permita a realização de diversos ensaios, tenha flexibilidade para alteração do modelo e ferramentas eficientes para análise dos resultados. Assim, é possível realizar testes com as variáveis de entrada, as técnicas de pré-processamento e a estrutura da rede para chegar no melhor modelo possível.

2.2 Redes Neurais

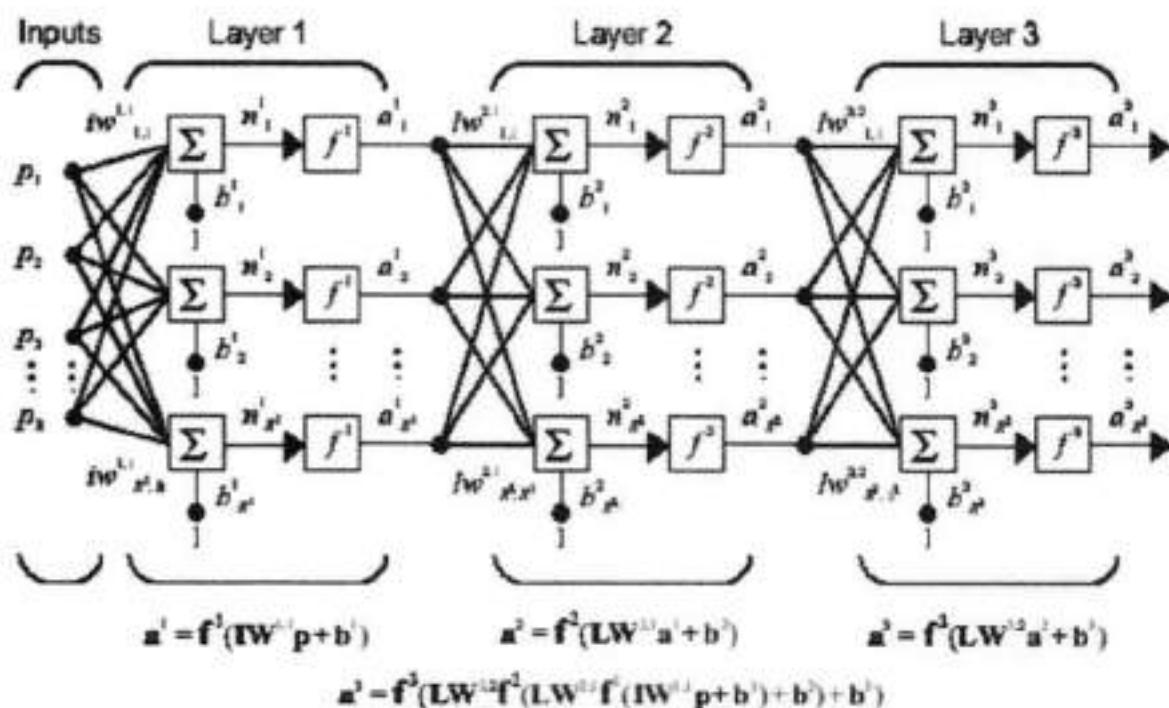
Redes neurais são inspiradas na forma em que o cérebro processa as informações e assim como no cérebro, que possui neurônios, as redes neurais são divididas em nós, e cada nó recebe diversas entradas, atribui pesos a essas entradas, e gera uma saída. De forma genérica, elas são uma ferramenta de modelagem matemática não-linear que podem ser entendidas como aproximadores universais. Evidentemente, assim como qualquer outra ferramenta de modelagem, elas possuem limitações, mas quando utilizadas da maneira correta, podem chegar a ótimos resultados.

Hoje em dia utiliza-se redes neurais em diversos campos, desde a biologia até a economia, em diversas aplicações, como reconhecimento de padrões, onde a rede obtém uma série de dados de entrada e decide a que grupo aquele conjunto de dados pertence. São utilizadas também para tomada de decisão, onde coloca-se os dados de entrada e a rede "responde" conforme os padrões observados. No caso deste trabalho, utilizamos as redes para obter estimativas de valores numéricos, não-linearmente relacionada com as entradas.

Tipicamente, a estrutura das redes neurais é composta por diversas camadas, e as saídas dos neurônios de uma camada são ligados à entrada de cada neurônio da próxima camada, chegando até a saída final. A saída de um nó é a aplicação de uma função sigmoidal (ou função com comportamento semelhante) na soma ponderada das entradas pelos seus pesos. Para atribuir os pesos corretos para cada entrada de cada nó é feito um processo de treinamento, ou aprendizado da rede neural. É nesse processo que se utiliza o que se tem de informação sobre o

fenômeno: os dados. Esses dados são colocados como entradas na rede neural e analisa-se o que se tem na saída, comparando o que deveria ser a saída idealmente. Assim, utiliza-se um algoritmo para ajustar os pesos dos nós de acordo com os dados até chegar aos melhores pesos, ou seja, que almejam o menor erro quadrático médio.

Figura 3 - Estrutura de uma rede neural de três camadas.



Fonte: Figura original da documentação da toolbox de redes neurais do MATLAB2011b.

Um dos algoritmos de otimização de pesos mais conhecidos é o *backpropagation*. Trata-se de um algoritmo que se baseia na ideia de adaptar os pesos no sentido negativo do gradiente do erro quadrático no espaço dos n pesos, de forma que o erro diminui o mais rapidamente possível. Isso garante que, desde que os pesos sejam adaptados a uma taxa não muito grande, o erro tenderá a atingir um mínimo local. Caso existam muitos mínimos locais, pode acontecer da rede neural obter resultados diferentes dependendo do ponto inicial onde os pesos começam a ser adaptados, ou seja, é necessário ter a capacidade de iniciar de

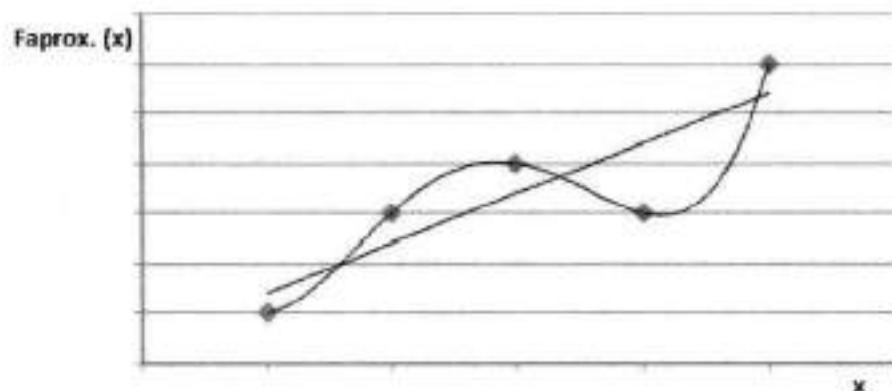
vários pontos diferentes de modo a não ficar sujeito a sempre cair no mesmo mínimo local (que pode ter um valor显著mente maior do que o mínimo global).

Utilizando esta lógica, é possível implementar um algoritmo iterativo em que o erro quadrático em relação a cada peso é obtido propagando o erro a partir da camada de saída até a camada de entrada, utilizando derivadas parciais do erro em relação a cada peso.

Tão importante quanto o algoritmo de otimização e estrutura da rede é a forma que os dados são representados. Este aspecto pode ser a diferença entre o sucesso e o fracasso de uma rede neural. De maneira geral, quanto mais se sabe sobre os detalhes do problema, menos a rede terá que "aprender" e melhor será a modelagem.

Antes de treinar a rede neural, é importante escolher uma quantidade conveniente de camadas e neurônios por camada. De forma análoga, a uma regressão polinomial, se utilizarmos um polinômio de grau alto para modelar um fenômeno que deveria ser uma reta, o polinômio vai passar por todos os pontos mas vai trazer resultados muito piores entre os pontos, como ilustra a figura abaixo. O mesmo ocorre com as redes neurais com excesso de parâmetros livres. Este problema é chamado de *overfitting*.

Figura 4 - Exemplo de um caso onde uma regressão polinomial com polinômio de grau muito alto prejudica o resultado.



Fonte: TRYPHONAS M. Modeling Hedge Fund Performance Using Neural Network Models, 2012, 63p.

2.3 Redes Neurais – Saídas

Como dito anteriormente, desejamos informar ao usuário do software FEPREM em que classe de ativo estão os recursos de um FIM.

A rentabilidade de um FIM é uma combinação não-linear da rentabilidade das classes de ativos, por isso nosso desejo de utilizar redes neurais. Abaixo serão detalhadas as classes de ativos³ que usamos.

1. IMA-B 5
2. IMA-B 5+
3. Taxa CDI
4. Dólar
5. Índice de Energia Elétrica IEE
6. Índice do Setor Industrial INDEX
7. Índice de Consumo ICON
8. Índice Financeiro IMOB
9. Índice Financeiro IFNC
10. Índice de Materiais Básicos IMAT
11. Índice Utilidade Pública UTIL

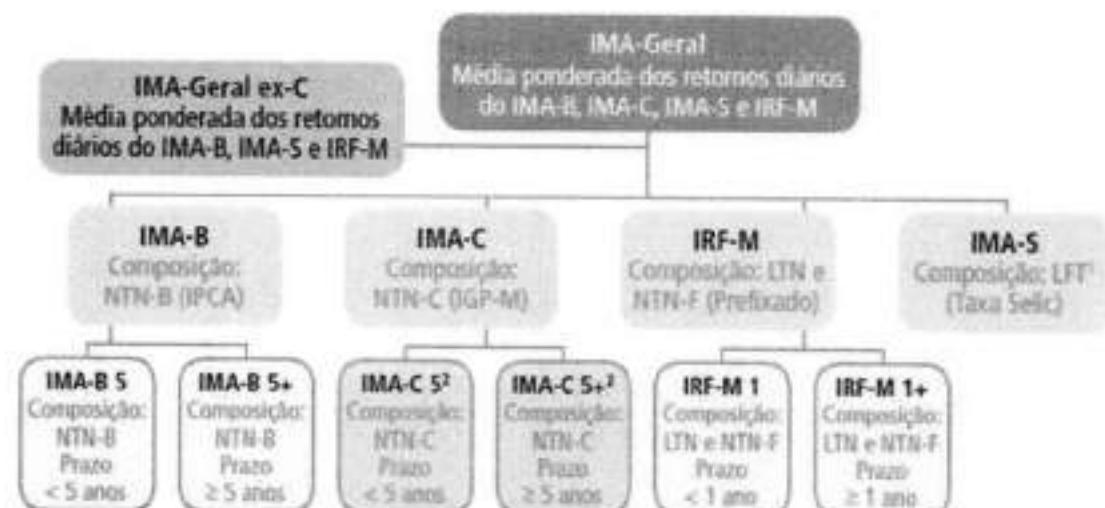
³ Rentabilidade das classes de ativos e dos FIM's são divulgadas pela Anbima e pela Bovespa.

Os três primeiros índices são de renda fixa. Os outros, de renda variável (ações). Vamos começar por detalhar os índices de renda fixa, seguido pelos de renda variável.

2.3.1 Renda fixa: IMA

O IMA, Índice de Mercado ANBIMA é uma família de índices que representa a evolução, a preços de mercado, da carteira de títulos públicos e serve como benchmark para o segmento. Com o objetivo de atender às necessidades dos diversos tipos de investidores e das suas respectivas carteiras, o IMA é atualmente subdividido em quatro sub índices, de acordo com os indexadores dos

Figura 5 - Detalhamento das classes de ativos de renda fixo



Fonte: ANBIMA

títulos: prefixados, indexados ao IPCA, indexados ao IGP-M e pós-fixados (Taxa Selic). São calculados sub índices com base nos prazos de seus componentes.

Dos índices de renda fixa, iremos optar por não utilizar o IMA-C, dada a baixa liquidez. Iremos assumir inicialmente que os FIM's não aplicam nesses ativos. Para efeito de índice pós-fixado, utilizaremos a taxa CDI, que é o índice atrelado a LFT.

Segue uma descrição dos títulos dos títulos públicos formadores dos índices acima:

A LTN é um título prefixado, o que significa que possui rentabilidade definida no momento da compra. Esse título possui fluxo de pagamento simples, ou seja, o investidor faz a aplicação e recebe o valor de face (valor investido somado à rentabilidade), na data de vencimento do título.

A NTN-F também é um título prefixado, com rentabilidade definida no momento da compra. Porém, diferentemente da LTN, seu rendimento é recebido pelo investidor ao longo do investimento por meio de cupons semestrais de juros, e na data de vencimento do título, quando do resgate do valor de face (valor investido somado à rentabilidade) e pagamento do último cupom de juros. O fluxo de cupons semestrais de juros aumenta a liquidez, possibilitando reinvestimentos.

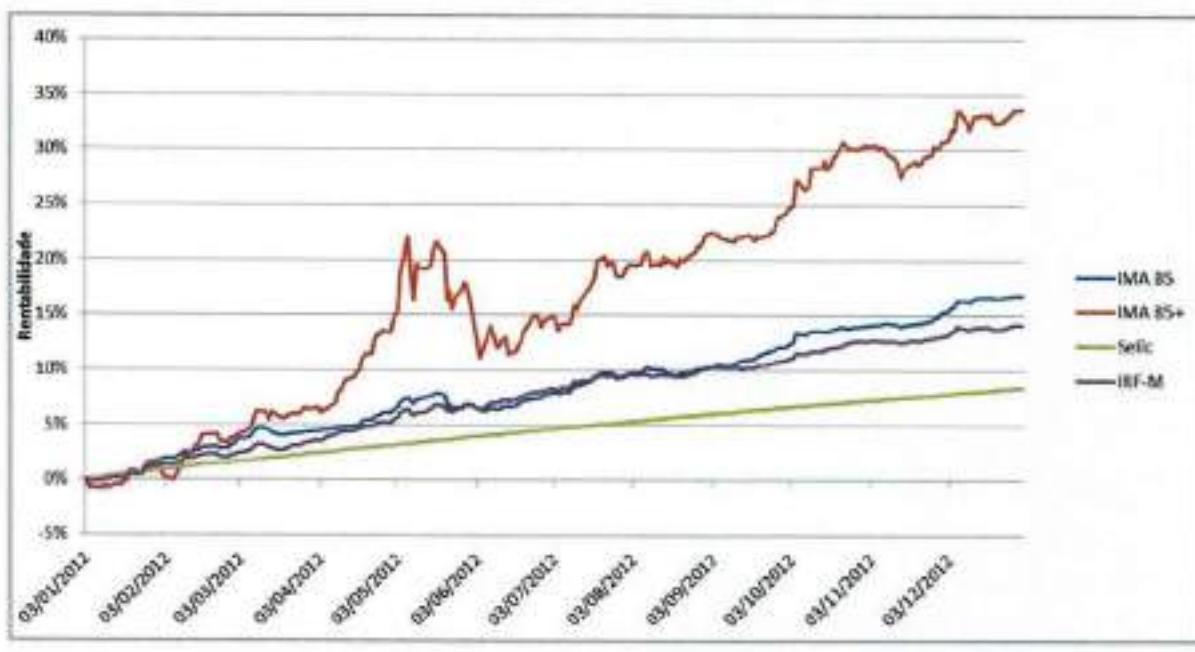
A NTN-B é um título com rentabilidade vinculada à variação do IPCA, acrescida dos juros definidos no momento da compra. Esse título permite ao investidor obter rentabilidade em termos reais, mantendo seu poder de compra ao se proteger de flutuações do IPCA durante a aplicação. Apesar de ser o título que possui o maior prazo para aplicação (atualmente conta com investimentos até 2045) seu rendimento é recebido pelo investidor ao longo do investimento, por meio de cupons semestrais de juros, e na data de vencimento do título, quando do resgate do valor de face (valor investido somado à rentabilidade) e pagamento do último cupom de juros.

A LFT é um título pós-fixado cuja rentabilidade segue a variação da taxa Selic, a taxa de juros básica da economia. Sua remuneração é dada pela variação da taxa Selic diária registrada entre a data de liquidação da compra e a data de vencimento de título, acrescida, se houver, de ágio ou deságio no momento da compra. A LFT é indicada para o investidor que deseja rentabilidade pós-fixada indexada à taxa de juros da economia (Selic). Além disso, o valor de mercado da LFT apresenta baixa volatilidade, evitando perdas no caso de venda antecipada. Por isso também, sua rentabilidade tende a ser mais baixa que a dos demais títulos. A LFT possui fluxo de pagamento simples, ou seja, o investidor faz a aplicação e recebe o

valor de face (valor investido somado à rentabilidade) na data de vencimento do título.

Como podemos ver no gráfico abaixo, títulos de renda fixa também possuem risco, em específico os prefixados: NTN-B, NTN-F e LTN. Essa volatilidade ocorre devido à expectativa dos juros básicos da economia. Conforme a expectativa de juros sobe, pede-se uma taxa indexada maior. Exemplificando: caso hoje decida-se comprar uma NTN-B com vencimento em 2015, o juro negociado será em torno de 2,90%. Supondo que dentro de um mês o Copom decide aumentar a taxa básica de juros, Selic, teremos então que a mesma NTN-B será negociada agora por uma taxa maior do que 2,90%, portanto o título perdeu valor.

Figura 6 - Rentabilidade de classes de ativos de renda fixa em 2012



Fonte: Autor

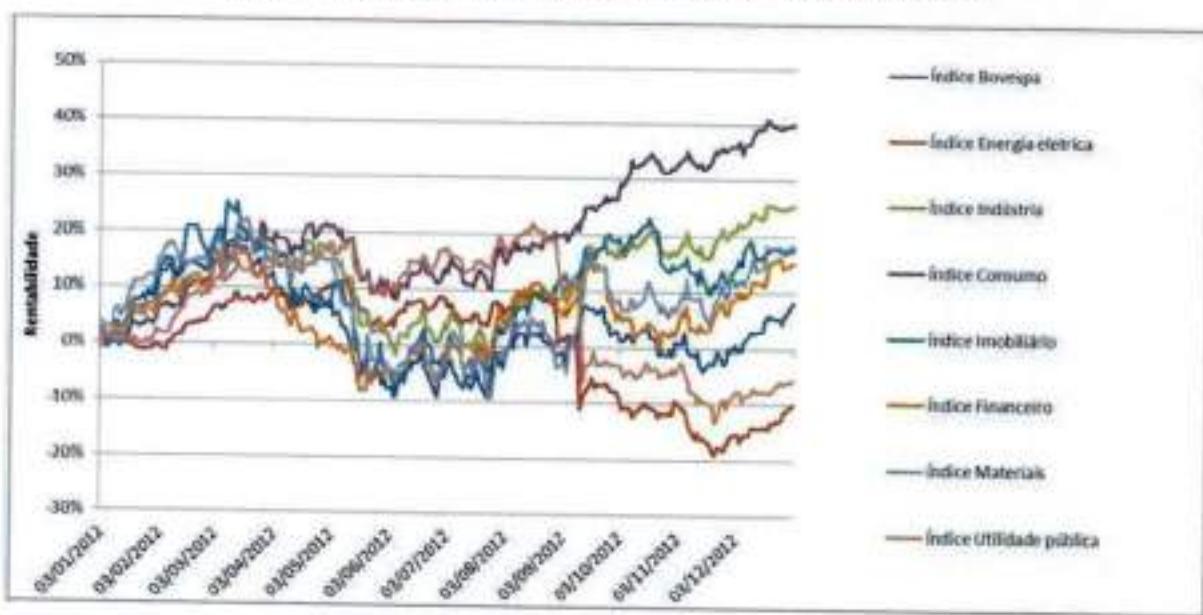
2.3.2 Renda variável

Não iremos utilizar o Índice Ibovespa para modelar os fundos multimercado, pois a maioria dos gestores faz investimentos em ações com base em uma análise setorial da economia do país, de forma que a modelagem por índice setorial é muito mais interessante. No ano de 2012, o Índice Ibovespa obteve rentabilidade de 7,40% enquanto que a composição de uma carteira com ações de empresas do setor de

consumo teve rentabilidade de 40,40% no ano. Muitos gestores investiram nesse setor em detrimento a setores como o de energia elétrica, que caiu cerca de 10%.

É de interesse de nosso projeto informar ao nosso cliente o quanto do dinheiro investido dele está em cada setor da economia. Dessa forma o estrategista do cliente poderá realocar os recursos do cliente visando uma exposição maior ou menor em determinado setor da economia, de acordo com suas expectativas de retorno.

Figura 7 - Rentabilidade de classes de ativos de renda variável em 2012



Fonte: Autor

Há anos presentes nos mercados internacionais, os índices setoriais têm o objetivo de oferecer uma visão segmentada do comportamento dos mercados de ações. Eles são constituidos pelas empresas abertas mais significativas de setores específicos, representando uma medida de comportamento agregado do segmento econômico considerado. Abaixo a descrição dos índices citados anteriormente.

Índice de Energia Elétrica IEE: Primeiro índice setorial da BM&FBOVESPA, o Índice de Energia Elétrica (IEE) foi lançado em agosto de 1996 com o objetivo de medir o desempenho do setor de energia elétrica. Dessa forma constitui-se em um

instrumento que permite a avaliação da performance de carteiras especializadas nesse setor.

Índice do Setor Industrial INDX: Fruto de um convênio entre as FIESP e a BM&FBOVESPA, o Índice do Setor Industrial (INDX) foi desenvolvido com o objetivo de medir o desempenho das ações mais representativas do setor industrial, importante segmento da economia brasileira. Sua carteira teórica é composta pelas ações mais representativas da indústria, que são selecionadas entre as mais negociadas na BM&FBOVESPA em termos de liquidez e são ponderadas na carteira pelo valor de mercado das ações disponíveis à negociação.

Índice de Consumo ICON: O Índice BM&FBOVESPA de Consumo (ICON) tem por objetivo oferecer uma visão segmentada do mercado acionário, medindo o comportamento das ações das empresas representativas dos setores de consumo cíclico e não-cíclico. As ações componentes são selecionadas por sua liquidez, e são ponderadas nas carteiras pelo valor de mercado das ações disponíveis à negociação.

Índice Imobiliário IMOB: O Índice BM&FBOVESPA Imobiliário (IMOB) tem por objetivo oferecer uma visão segmentada do mercado acionário, medindo o comportamento das ações das empresas representativas dos setores de atividade imobiliária compreendidos por construção civil, intermediação imobiliária e exploração de imóveis. As ações componentes são selecionadas por sua liquidez, e são ponderadas nas carteiras pelo valor de mercado das ações disponíveis à negociação.

Índice Financeiro IFNC: O Índice BM&FBOVESPA Financeiro (IFNC) tem por objetivo oferecer uma visão segmentada do mercado acionário, medindo o comportamento das ações das empresas representativas dos setores de intermediários financeiros, serviços financeiros diversos, previdências e seguros. As ações componentes são selecionadas por sua liquidez, e são ponderadas nas carteiras pelo valor de mercado das ações disponíveis à negociação.

Índice de Materiais Básicos IMAT: O Índice de Materiais Básicos (IMAT) BM&FBOVESPA tem por objetivo oferecer uma visão segmentada do mercado

acionário, medindo o comportamento das ações das empresas representativas do setor de Materiais Básicos. As ações componentes são selecionadas por sua liquidez e ponderadas nas carteiras pelo valor de mercado das ações disponíveis à negociação.

Índice de Utilidade Pública UTIL: O Índice de Utilidade Pública (Util) BM&FBOVESPA tem por objetivo oferecer uma visão segmentada do mercado acionário, medindo o comportamento das ações das empresas representativas do setor de utilidade pública (energia elétrica, água e saneamento e gás). As ações componentes são selecionadas por sua liquidez e ponderadas nas carteiras pelo valor de mercado das ações disponíveis à negociações.

2.4 Detalhamento do Problema

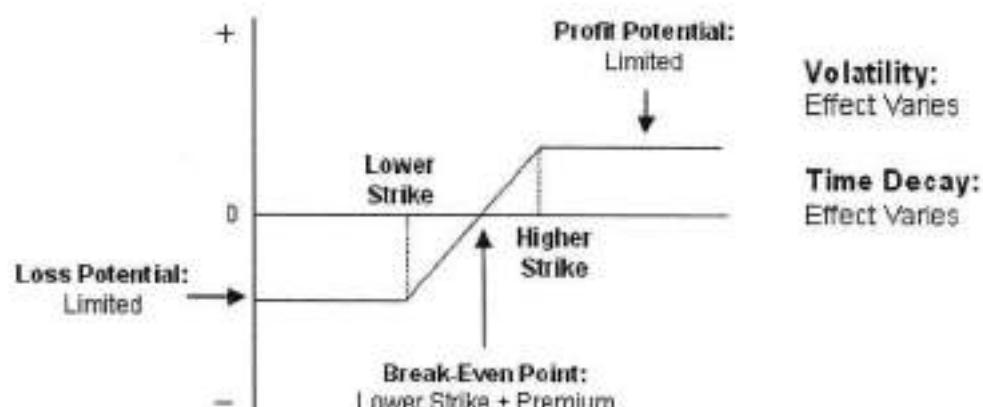
Uma vez conhecidas as classes de ativos de renda fixa e variável, vamos voltar ao problema que desejamos solucionar. Podemos dizer que um fundo possui recursos nas classes de ativos IMA-B 5 e Dólar, se por exemplo ele possuir uma NTN-B com vencimento em 2015 e ações do Banco Itaú. Nesse simples caso, a rentabilidade do FIM será uma combinação linear das rentabilidades das classes de ativos IMA-B 5 e Dólar. Iremos apresentar a exposição em dez classes de ativos, a classe CDI não é mostrada pois não apresenta risco ao investidor.

Para entender uma parte da não-linearidade do nosso problema é necessário saber como funciona o mercado de opções, explicado sucintamente abaixo.

O Mercado de Opções é o mercado em que são negociados direitos de compra ou venda de um lote de ações, com preços e prazos de exercício preestabelecidos. Esse mercado foi criado com o objetivo básico de oferecer um mecanismo de proteção ao mercado de ações contra possíveis perdas. Uma vez que os preços e retornos dos instrumentos financeiros estão sujeitos a flutuações imprevisíveis. Uma opção é o direito de comprar ou vender um ativo específico, por um preço, adquirido mediante o pagamento de um valor (prêmio), para ser exercício em uma data preestabelecida (data de vencimento).

Uma estratégia comum adotada, chamada de *Call Spread*, é detalhada abaixo. Se o investidor compra o call spread, isso significa que o ativo-objeto (no nosso caso ação do Itaú) subir, o investidor ganha a alta do índice até certo patamar percentual de alta, por exemplo 20%. Porém, se o ativo subir mais do que 20%, o cliente só ganha 20% no período da operação. Por outro lado, se o ativo cair, o cliente tem o capital garantido.

Figura 8 - Gráfico detalhando a não linearidade da rentabilidade do mercado de opções.



Fonte:.cboe.com

Não é objeto de estudo desse trabalho nos aprofundar na estruturação desse produto, mas sim saber que existem aplicações desse tipo e que elas possuem retorno não lineares, fato que reforça o desejo de utilizar uma rede neural como solução.

A não linearidade também está presente nas decisões de compra e venda de ativos, como veremos no desenvolvimento.

Os fundos que modelamos são os da categoria Multimercado isso possibilita uma abertura legal para que o fundo esteja exposto mais de um patrimônio, conceito que recebe o nome de alavancagem.

2.4.1 Alavancagem

"A amplitude dos mercados de derivativos aliada a algumas das formas contemporâneas de gestão financeira (em particular de certos tipos de fundos de investimento e de carteiras administradas) trazem à tona outra questão que encerra importantes repercussões macroeconómicas. Trata-se da possibilidade de estabelecimento de diferentes níveis de especulação que variam na função direta do grau de alavancagem das carteiras" (FARHI MARYSE, 1999, p.13)

Alguns dos fundos multimercado que modelamos se alavancam com objetivo de obter maiores rentabilidades com operações em certos ativos. Exemplificando: se o gestor está convicto que os juros básicos (SELIC) irão cair, ele poderá montar uma posição alavancada em juros-pré (classe que se valoriza com a queda da SELIC), dessa forma o fundo poderá por exemplo aplicar 120% do patrimônio do fundo em juros-pré e continuar aplicado em outras classes de ativo consequentemente a soma das exposições por classe de ativo somam mais de 100% (fato que ocorreu com bastante frequência em nossos ensaios). É importante ressaltar que esse tipo de operação poderá resultar perdas superiores ao patrimônio do fundo, exigindo aportes adicionais dos cotistas para honrar os contratos derivativos firmados.

2.5 Definições gerais sobre a estrutura da plataforma

Iremos agora realizar uma análise geral sobre as alternativas de concepção da plataforma que desejamos desenvolver. Ficou definido que iria tratar-se de uma plataforma web, rodando diretamente do navegador do usuário, sem nenhum tipo de download. Esta definição inicial se fez necessária pois permite que o código do programa fique protegido em um servidor remoto. Para que pudéssemos guiar e limitar nosso escopo, definiu-se que esta plataforma será desenvolvida com o

padrão de design de software denominado *Model-View-Controller*, ou simplesmente MVC.

Por que utilizar MVC? Porque é um padrão de projeto de software extensivamente testado que transforma uma aplicação em pacotes de desenvolvimento rápido, modular e de fácil manutenção. Novas funcionalidades são facilmente adicionadas e pode-se dar nova característica à partes específicas da aplicação de maneira ágil. Tem design modular e separado que também permite aos desenvolvedores e designers trabalharem simultaneamente, incluindo a capacidade de se construir um protótipo muito rapidamente. A separação também permite que os desenvolvedores alterem uma parte da aplicação sem afetar outras.

Ao programar em MVC, separa-se uma aplicação em três partes principais: *Model*, *View* e *Controller*.

2.5.1 A camada *Model*

A camada *Model* repousa acima dos dados a serem utilizados no projeto. É responsável pela obtenção de dados e conversão destes dados em conceitos significativos para o projeto em questão. Dentre estas atividades, podemos destacar validação (verificação da integridade dos dados), pré-processamentos, associação e outras tarefas.

2.5.2 A camada *View*

Trata-se da camada de apresentação. Manipula o que será visualizado pelo usuário final, de acordo com os dados disponibilizados pela camada *Model*. É nesta camada que estará armazenada toda a interface do programa.

2.5.3 A camada Controller

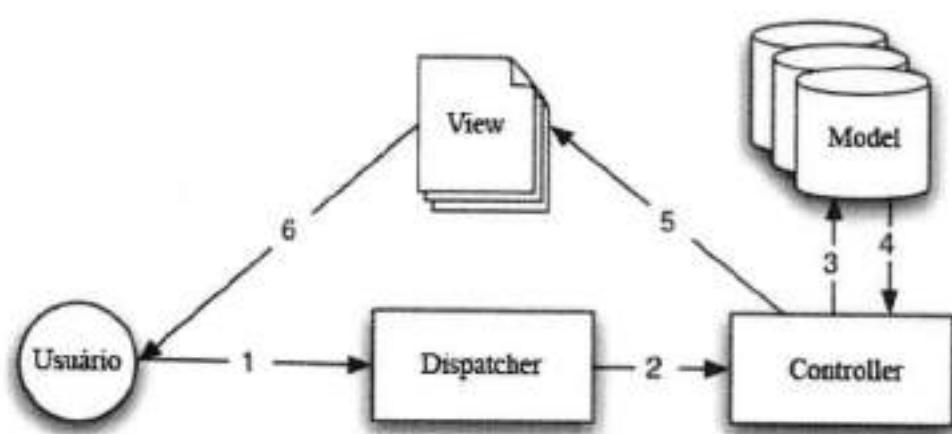
A camada *Controller* (controlador) lida com as requisições dos usuários. É responsável por retornar uma resposta com a ajuda das camadas *Model* e *View*.

Os *Controllers* podem ser vistos como gerentes tomando os devidos cuidados para que todos os recursos necessários para completar uma tarefa sejam delegados para os trabalhadores corretos. Ele aguarda os pedidos dos usuários, verifica a validade de acordo com as regras de autenticação e autorização, delega dados para serem obtidos ou processados pelos *Models* e seleciona o tipo correto de apresentação dos dados para finalmente delegar o trabalho de renderização para a camada *View*.

2.5.4 Ciclo de requisição da arquitetura MVC

Para que haja melhor entendimento do funcionamento da arquitetura MVC, apresentaremos um ciclo de requisição de forma mais detalhada.

Figura 9 - Esquema de ciclo de requisição da arquitetura MVC



Fonte: book.cakephp.org

Um ciclo de requisição típico da arquitetura MVC começa com o usuário solicitando um recurso em sua aplicação. Esta solicitação é primeiramente processada por um *dispatcher* (expedidor) que irá selecionar o objeto *Controller* correto para lidar com a solicitação feita.

Assim que a solicitação do cliente chega ao *Controller*, este irá se comunicar com a camada *Model* para processar qualquer operação de busca ou armazenamento de dados que for necessário. Após esta comunicação terminar, o *Controller* continuará delegando, agora para o objeto *View* correto a tarefa de gerar uma saída resultante dos dados fornecidos pelo *Model*.

Finalmente quando a saída é gerada, ela é imediatamente enviada para o usuário.

2.5.5 Alternativas de linguagens de programação

A escolha da linguagem de programação a ser utilizada é um dos itens fundamentais do processo de desenvolvimento da plataforma. Esta decisão afeta o projeto de diversas maneiras, como a procura por profissionais capacitados ou mesmo o próprio desempenho de nossa aplicação.

Primeiramente será dado um resumo das características de algumas das linguagens candidatas, apontando sempre que possível seus pontos fortes e fracos.

2.5.5.1 PHP

PHP é uma linguagem server-side criada principalmente para desenvolvimento de aplicações web. Atualmente, está instalada em mais de 200 milhões de websites e 2 milhões de servidores. Originalmente criado por Rasmus Lerdorf, PHP significa PHP: *Hypertext Preprocessor*.

O código PHP é interpretado por um servidor web com um módulo de processamento que gera os resultados de uma página web. Os comandos PHP podem estar dentro de uma página HTML ou em um script próprio. Trata-se de um software livre, lançado sob uma licença denominada PHP License.

2.5.5.2 C++

C++ é uma linguagem de programação multi-paradigma e de uso geral. A linguagem é considerada de médio nível, pois combina características de linguagens de alto e baixo níveis. Desde os anos 1990 é uma das linguagens comerciais mais populares, sendo bastante usada por seu grande desempenho e comunidade de usuários.

2.5.5.3 Matlab

Matlab, que significa *Matrix Laboratory*, é um ambiente de computação numérica da quarta geração de linguagens de programação. Desenvolvido pela MathWorks, MATLAB permite manipulação de matrizes, plotagem de funções, implementação de algoritmos, criação de interfaces de usuário além de interfaces de comunicação com outras linguagens de programação, como por exemplo C, C++, Java ou PHP.

2.5.5.4 VBA

Visual Basic for Applications (VBA) é uma implementação da linguagem de programação Visual Basic 6 da Microsoft e sua IDE (*intregrated development enviroment* ou ambiente de desenvolvimento integrado).

VBA permite a construção de funções definidas pelo usuário, automação de processos além de outros procedimentos de baixo nível. Trata-se de uma extensão das chamadas macros, muito comuns em programas como o Excel.

2.5.5.5 Python

Python é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem como um todo não é formalmente especificada.

Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.

2.5.5.6 Alternativa escolhida

A escolha inicial será pela linguagem de programação PHP. Dentre os motivos estão a facilidade de aprendizado, popularidade, grande comunidade de desenvolvedores ativa, e familiaridade. Há também facilidade de integração com outras linguagens, como por exemplo VBA (através de planilhas em Excel) ou Matlab.

2.5.6 Alternativas para armazenamento de dados

Além das escolhas de metodologia de design de software e linguagem de programação, é importante também selecionar uma solução responsável pelo armazenamento de dados. Analisaremos agora as alternativas mais populares.

2.5.6.1 PostgreSQL

Trata-se de um sistema de banco de dados *open-source* e ORM (*Object-Relational Mapping*). Possui estrutura robusta e confiável, rodando em grande parte dos sistemas operacionais mais populares, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e Windows.

Dentre suas características principais podemos apontar: suporte total a *foreign keys, joins, views, triggers e stored procedures*. Possui a maior parte dos tipos de dado do SQL:2008, incluindo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL e TIMESTAMP.

2.5.6.2 SQLite

SQLite é uma biblioteca de implementação de bancos de dados SQL construída na linguagem C que apresenta como algumas de suas principais características a não-necessidade de um servidor e a ausência de pré-configuração. Seus arquivos de bancos de dados não são dependentes da plataforma utilizada. É voltado para aplicações onde a simplicidade é mais importante do que recursos extras de alto nível.

2.5.6.3 MySQL

MySQL é um dos sistemas *open source* de gerenciamentos de bancos de dados relacionais mais populares do mundo. Suas aplicações são interpretadas em um servidor, permitindo acesso de múltiplos usuários a diversos bancos de dados. Trata-se de uma escolha popular quando se trata de aplicações web.

2.5.6.4 Alternativa escolhida

Dentre as opções acima apresentadas, a solução escolhida (ao menos em caráter temporário) é o MySQL. Isto por que é um sistema relativamente seguro e bastante popular – o que se traduz em grande comunidade de usuários. Além disso, combina perfeitamente com a linguagem de programação PHP quando se deseja criar aplicações web.

2.5.7 Interface Gráfica

Como foi mencionado anteriormente, o projeto em questão será confeccionado inteiramente como uma plataforma web, sem nenhum tipo de download necessário para o usuário. Dada esta importante premissa, é fácil definir como será construída a interface gráfica de software: através das linguagens de programação HTML e CSS. É através da junção destes dois itens que a chamada View do projeto é definida. Segue abaixo uma explicação resumida sobre cada um destes itens.

2.5.7.1 HTML

O termo HTML significa *HyperText Markup Language* (que significa Linguagem de Marcação de Hipertexto). Trata-se uma linguagem utilizada para construção de páginas web. Seus scripts são interpretados pelos navegadores, que, com base em suas marcações, definem o que será apresentado na página.

2.5.7.2 CSS

O termo CSS significa (*Cascade Styling Sheets*) define uma linguagem de programação voltada somente para o design de páginas WEB. Sua característica

mais apreciada está ligada à separação entre o que é apresentação e o que é formatação de conteúdo.

3 ANÁLISE DE VIABILIDADE

Podemos avaliar a viabilidade de nosso projeto em três frentes: viabilidade técnica de modelagem matemática e viabilidade técnica de software e viabilidade econômica.

3.1 Modelagem Matemática

Sob o ponto de vista de modelagem matemática, sabemos que um projeto similar ao nosso já foi desenvolvido em uma tese de mestrado da Universidade de Toronto. Segundo o trabalho, os resultados obtidos foram bons para mais de 200 FIM's. O projeto lá desenvolvido é apenas uma parte do nosso projeto, pois nós desenvolvemos um produto que possa ser comercializado, não apenas elaboramos o modelo para uso próprio.

Por nosso projeto ser mais abrangente, fizemos algumas simplificações iniciais, como: inicialmente criar um FIM fictício com certas regras de investimento e desinvestimento segundo as rentabilidades de classes de ativos. Inicialmente incluímos poucos produtos de rentabilidade não linear na carteira do fundo, e na medida que viamos quais os melhores parâmetros para alterar na rede neural, melhor ficava nossa previsão, podendo então utilizar nossa rede para atuar em casos reais com mais não-linearidades.

Levando em consideração o raciocínio acima, ficamos motivados a executar o projeto pelas ótimas chances que existiam de produzir bons resultados iniciais quanto a modelagem do FIM. Contudo nosso projeto não se resume a essa modelagem. Além disso, realizamos uma série de medidas de erro dos valores de saída e, por último, nosso programa também fornece a rentabilidade acumulada de diversos fundos de investimento multimercado. Todas essas ferramentas foram obtidas com um bom desenvolvimento de arquitetura de nosso software em conjunto com programação em diversas linguagens.

3.2 Viabilidade de Software

Quanto ao software, a parte que mais traz riscos do ponto de vista de sucesso do produto já em uso é a capacidade de processamento da grande quantidade de dados, já que tratando-se de um produto em plataforma web, ele deve ser capaz de receber múltiplos acessos simultâneos sem perder qualidade. Para a fase de desenvolvimento, esse problema não foi crítico, mas conforme exista uma maior demanda por usuários, deverá haver investimento em aumento de poder de processamento. Desde o início, a preocupação com a eficiência na criação do código e na elaboração da arquitetura existiu para que os problemas citados anteriormente fossem controlados. Os maiores desafios que o grupo enfrentou sob esse ponto de vista foram o estudo e a avaliação da qualidade do código do produto, prevendo eventuais problemas de eficiência. No entanto, o produto poderá passar por constantes evoluções e melhorias, visando manter a qualidade.

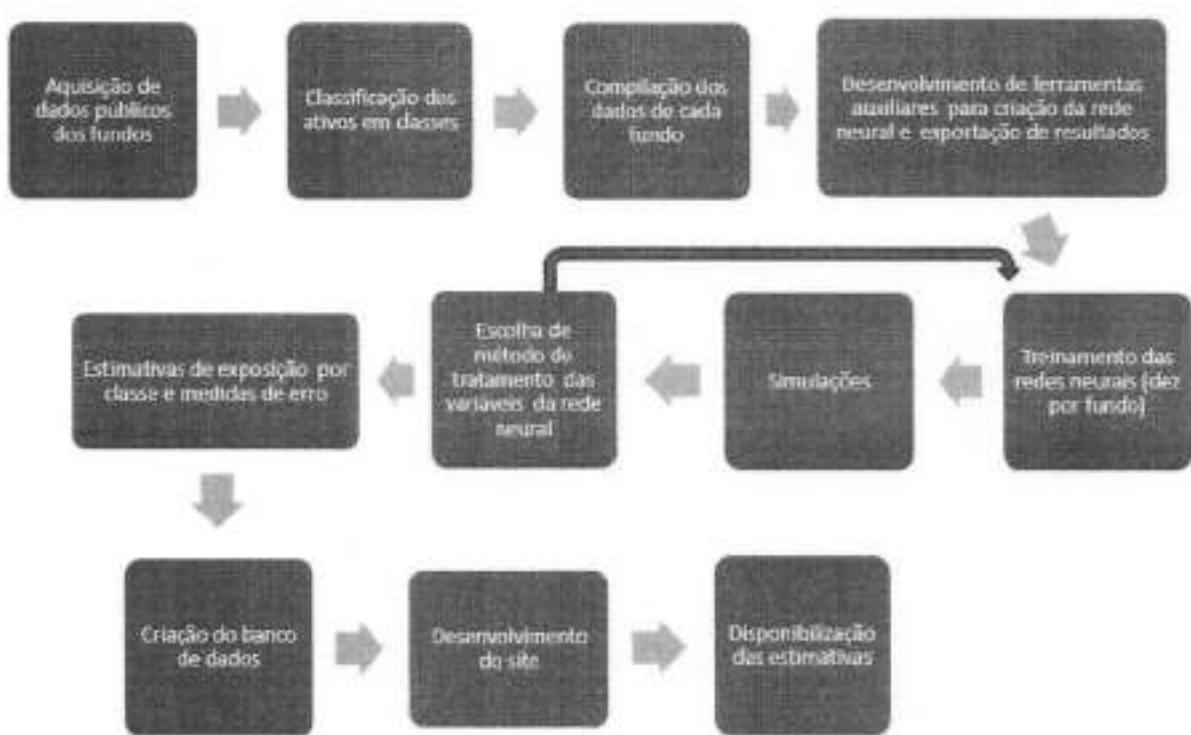
3.3 Viabilidade Econômica

Nosso produto final é utilizado via web. Nossos clientes (gestores) poderão utilizar o software de qualquer lugar, sem a necessidade de um conjunto de arquivos ou banco de dados instalado em seu computador. Tal fator torna os custos de produção e manutenção mais baratos. Todo controle de acesso, assim como atualizações de informações de bancos de dados poderá ser feito a distância.

3.4 Diagramas de projeto e produto

Temos a seguir o diagrama de blocos do nosso projeto, no desenvolvimento desse trabalho será detalhado cada uma das etapas.

Figura 10 - Fluxo do Projeto - FEPREM



Fonte: Autor

Partimos da aquisição de dados dos fundos, onde coletamos as informações sobre em que ativos cada fundo estava investido até seis meses atrás. Depois disso, é necessário classificar todos os investimentos entre as 10 classes que são analisadas. É necessário também a coleta de informações sobre a cota dos fundos e a rentabilidade de cada uma das dez classes de ativos.

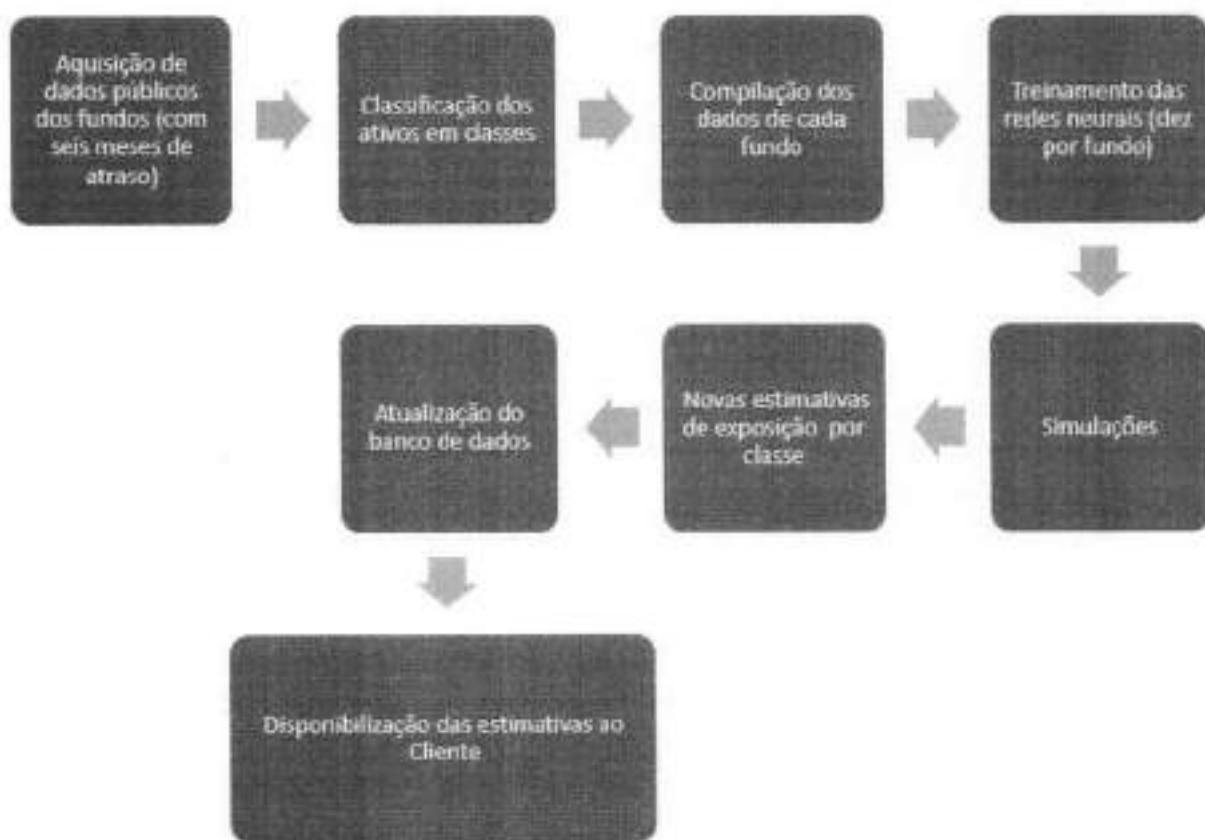
Após termos os dados de todos os fundos devidamente coletados e classificados, desenvolvemos ferramentas auxiliares para criação e análise das redes neurais, possibilitando escolher se os dados de entrada e saída são normalizados, qual dos quatro modelos de redes neurais é utilizado (no desenvolvimento do trabalho serão explicitados quais são os modelos), e quais seriam os valores dos parâmetros de treinamento das redes neurais, como passo e número de iterações. Além disso, as ferramentas possibilitam a gravação automática dos resultados em Excel, com gráficos, resultados e medidas de erro, tanto para o modelo neural quanto para o modelo linear.

São necessários diversos testes para encontrar as melhores condições para a previsão da rede neural. Após as condições serem encontradas, a simulação é realizada e os resultados são armazenados.

Tendo os resultados dos testes, eles são armazenados em um banco de dados, de forma a ficar disponível para ser usado pelo site. Com os dados disponíveis, desenvolvemos o site, disponibilizando as informações de forma clara e interessante para o usuário, com diversos gráficos.

Sob o ponto de vista de produto, temos um fluxo operacional relativamente mais simples, que pode ser representado pelo diagrama de blocos abaixo:

Figura 11 - Fluxo operacional do produto final



Fonte: Autor

Este seria o diagrama de blocos para o nosso projeto já em regime, ou seja após a realização de todas as etapas iniciais de preparação de software do site e desenvolvimento das redes neurais para cada classe de fundo.

Devemos ressaltar também que para este trabalho, abrangemos sete fundos multimercado reais além do fundo fictício, número que gerou a necessidade do treinamento de setenta redes neurais. No prosseguimento de desenvolvimento do produto, iremos modelar mais fundos.

4 CRIAÇÃO DO FUNDO FICTÍCIO

4.1 Motivação

A criação de um modelo baseado em redes neurais requer uma série de dados para realização de ensaios. A partir dessas informações pode-se variar diversos parâmetros, de forma a chegar nos melhores modelos. Com o objetivo de, num momento inicial, facilitar a criação do modelo, foi criado um fundo multimercado fictício, o *Simula Fundo de Investimento Multimercado*.

Tendo um fundo criado a partir de um algoritmo conhecido, que se comporta de forma parecida com um fundo real, podemos captar quais as variáveis mais importantes e qual o melhor pré-processamento dos dados de entrada, para que a rede neural funcione com desempenho adequado. A ideia é partir deste modelo, capaz de prever o comportamento do fundo fictício, adaptá-lo para resolver o problema real.

A grande vantagem de modelar a partir do fundo fictício é que sabemos a priori quais os dados que possuem correlação com a saída, temos uma quantidade maior de dados para treinar a rede neural. Além disso, sabemos a concentração por classe de ativo em todas as semanas, que é a informação que desejamos descobrir para os fundos reais com a diferença que para estes a informação é mensal. Criamos então a rede neural e treinamos para este fundo fictício com objetivo de estimar a concentração por classe de ativos, semana a semana.

4.2 Algoritmo de criação

O fundo foi simulado por um algoritmo programado em VBA, que lê as entradas necessárias no Excel e executa as compras e vendas a partir das regras pré-estabelecidas. Temos então as rentabilidades e exposições mês a mês do fundo.

O Fundo Simula foi criado com início em janeiro de 2008 e fim em outubro de 2012. Tal período foi escolhido de tal forma que todas as classes de ativos já

estivessem sendo apuradas. Durante esse período o fundo realiza compras e vendas de ativos semanalmente, para tanto, foi desenvolvido um algoritmo que simula a estratégia de um gestor. Seu princípio de funcionamento é simples porém bastante próximo da realidade e está detalhado abaixo.

Para Renda Fixa, semanalmente o fundo realiza compras e vendas de ativos conforme sua rentabilidade na semana, seguindo as formulas abaixo:

$|Rentabilidade\ do\ ativo| < 0,20\% \rightarrow \text{Não\ negocia}$

$0,20\% < |Rentabilidade\ do\ ativo| < 0,70\% \rightarrow \text{Compra}$

$|Rentabilidade\ do\ ativo| > 0,70\% \rightarrow \text{Venda}$

Para Renda Variável, a faixa de valores muda devido maior volatilidade dos preços. Abaixo as formulas:

$|Rentabilidade\ do\ ativo| < 1\% \rightarrow \text{Não\ negocia}$

$1\% < |Rentabilidade\ do\ ativo| < 3\% \rightarrow \text{Compra}$

$|Rentabilidade\ do\ ativo| > 3\% \rightarrow \text{Venda}$

Ao final de uma semana, temos todas as rentabilidades das classes de ativos (dados públicos).

No algoritmo, primeiro verifica-se os ativos que serão vendidos na semana, essa venda é parcial, uma fração do saldo atual naquele ativo. Realizada a venda, os recursos são somados ao valor financeiro em caixa (que é o dinheiro disponível para fazer investimentos). Esse procedimento é condizente com os fundos que não realizam alavancagem financeira, ou seja não estão expostos mais de um patrimônio, conforme veremos na seção de fundos reais, existem fundos que

alavancam através de instrumentos derivativos, para estes casos, foi necessário realizar algumas alterações no modelo de rede neural.

Uma vez analisados que ativos serão comprados e então utiliza-se do dinheiro em caixa para realizar as compras. Apenas em casos raros o caixa é zerado, em todas outras vezes cada classe de ativo tem uma parte fixa de recursos.

4.3 Comparação com fundos reais

A metodologia implementada que foi descrita acima possui o fluxo operacional semelhante a um fundo real que não realiza alavancagem, dessa forma garantimos que as rentabilidades calculadas possuam significado real, assim como a concentração por classe de ativos (objetivo da criação do Fundo Simula). A única variação no fluxo operacional, seria a possibilidade de alavancagem do fundo, o que no caso possibilita o investimento de mais de 100% do patrimônio. Se criássemos um fundo e seguissemos as ordens do algoritmo do Simula, teríamos as rentabilidades que são apresentadas abaixo.

No período de 03/01/2008 a 10/10/2012 o Fundo Simula rendeu 74%, enquanto que o CDI (parâmetro de comparação) rendeu 61%, ou seja, o Fundo Simula rendeu 124% do CDI, conforme gráfico comparativo abaixo.

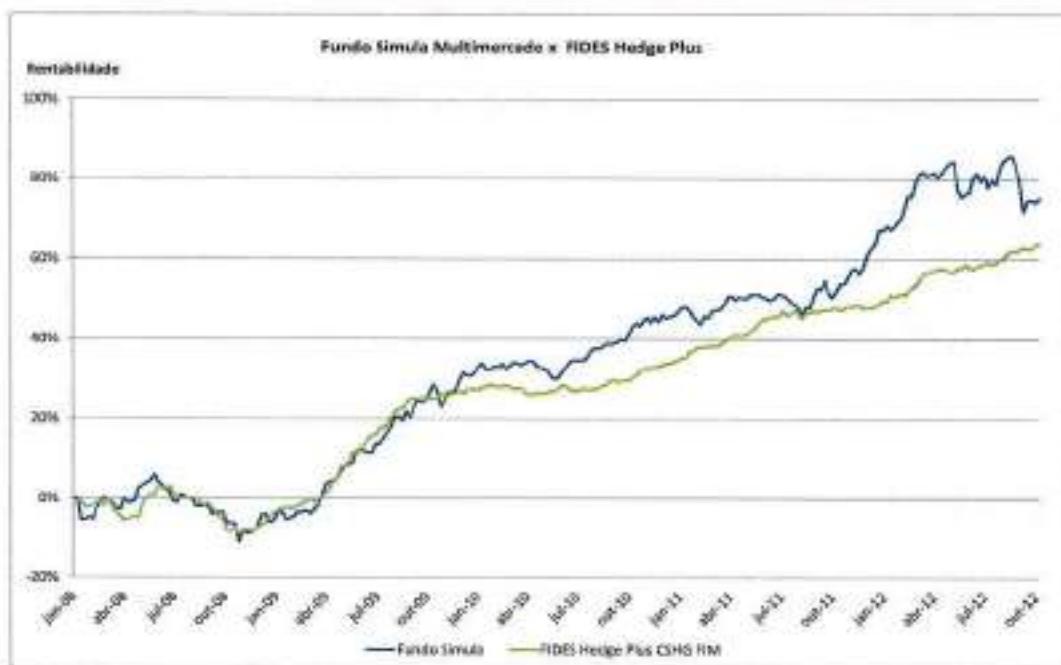
Figura 12 – Rentabilidade do fundo Simula em relação ao CDI



Fonte: Autor

Para efeito de comparação, temos o desempenho do Fundo Simula contra outro fundo multimercado, o FIDES Hedge Plus CSHG FIM. Como podemos observar no gráfico abaixo o Fundo Simula apresenta uma estratégia que gera rentabilidades próximas a de fundos multimercado reais e até superiores em um período recente.

Figura 13 - Rentabilidade do fundo Simula x Fundo FIDES Hedge Plus

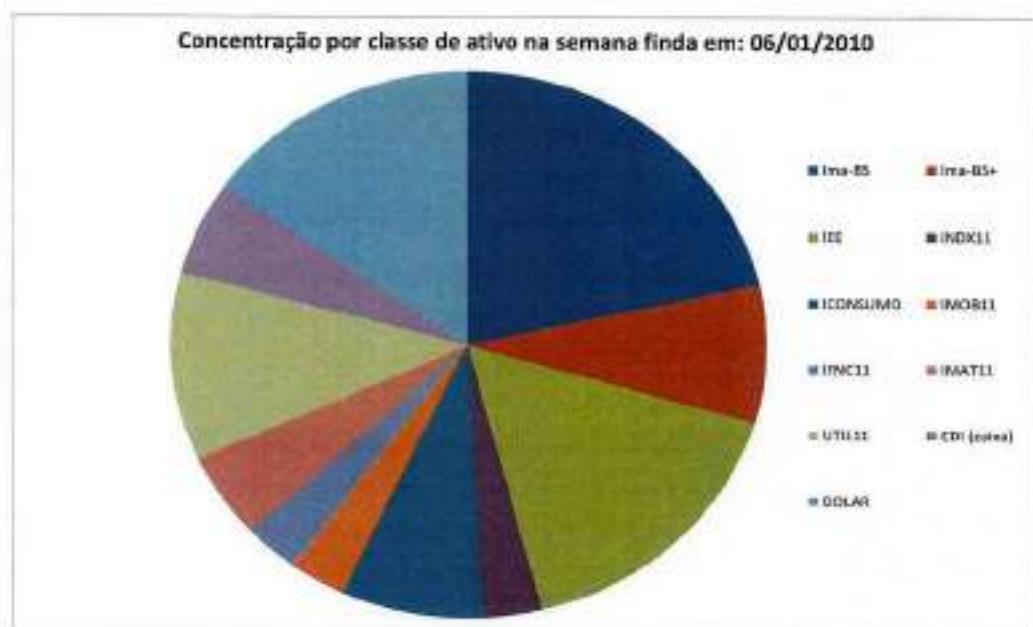


Fonte: Autor

Podemos concluir que o nosso fundo fictício é um bom ponto de partida para implementação da rede neural, pois apresenta rentabilidades similares a um fundo real além de um fluxo operacional muito semelhante.

Abaixo o gráfico da concentração média por classe de ativo do Fundo Simula na primeira semana de 2010.

Figura 14 - Concentração do fundo Simula em 06/01/2010



Fonte: Autor

4.4 Construção de modelos

Um dos maiores problemas encontrados ao se construir modelos sobre o mercado financeiro está na dificuldade de saber quais dados influenciam significativamente e quais não influenciam a saída. Os problemas geralmente são complexos e com muitas variáveis envolvidas e os dados disponíveis são limitados.

No caso do nosso problema em questão encontra-se as mesmas dificuldades. Sabe-se que a concentração por classe de ativo em um fundo é - em primeiro grau - função de:

- Concentrações de períodos anteriores;
- Rentabilidade de cada classe;
- Rentabilidade do fundo

Conforme mostrado empiricamente na seção anterior, além disso há um fator complicador: a quantidade de compra de ativos depende também da rentabilidade dos outros ativos. Exemplificando: digamos que em uma determinada semana o mercado evoluiu de forma favorável a compra para várias classes de ativos, como os

recursos do fundo são limitados o gestor irá comprar um pouco de cada classe de ativo e a variação de concentração será quase nula. Analisando o outro extremo: quase todas as classes evoluem de forma favorável a venda na semana e uma delas à compra, agora o gestor terá muitos recursos para fazer a compra, logo a concentração irá aumentar significativamente a variação será alta. Contudo esse fator complicador não é decisivo para os fundos que realizam alavancagem, uma vez que eles fazem uso de derivativos e se expõe mais de 100%, para eles a quantidade de recursos em caixa não é um fator limitador.

Podemos então concluir que há um fator – de segundo grau de influência – responsável pela variação de concentração de um ativo: A rentabilidade das classes dos outros ativos.

Conforme veremos a seguir, desenvolvemos dois modelos iniciais de rede neural, um que leva em consideração as variáveis de importância secundária e um outro que ignora essas variáveis.

4.4.1 Modelo 1 de rede neural – Fundo fictício

Neste modelo as variáveis de importância secundária não entram na rede neural. As variáveis de entradas da rede neural são:

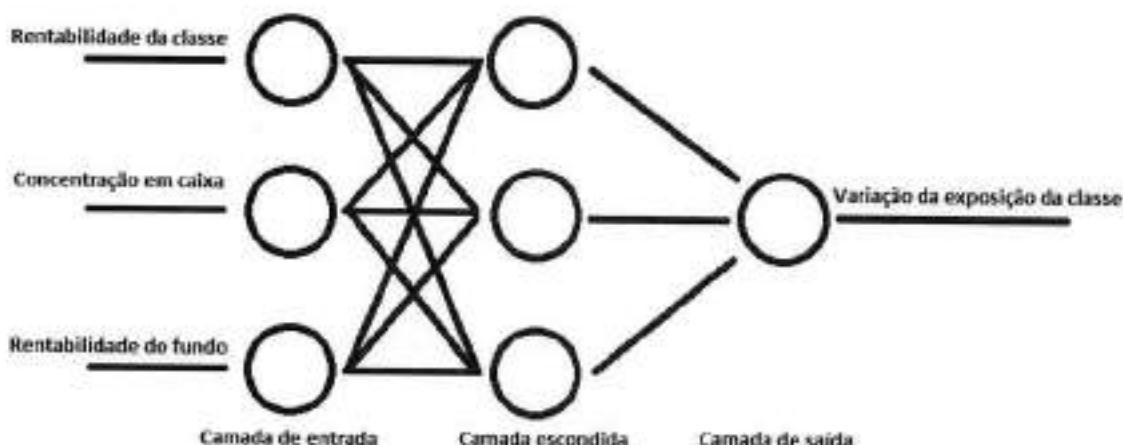
- Rentabilidade da classe de ativo na semana n,
- Concentração de caixa (que são os recursos disponíveis para aplicação) na semana n-1 e
- Rentabilidade da cota na semana n

E a saída é:

- Variação da concentração da classe de ativo.

Esse valor será aplicado a concentração da semana n-1 para obtermos a concentração na semana n. Este será nosso Modelo 1 de rede neural. Na figura abaixo temos o esquema da rede.

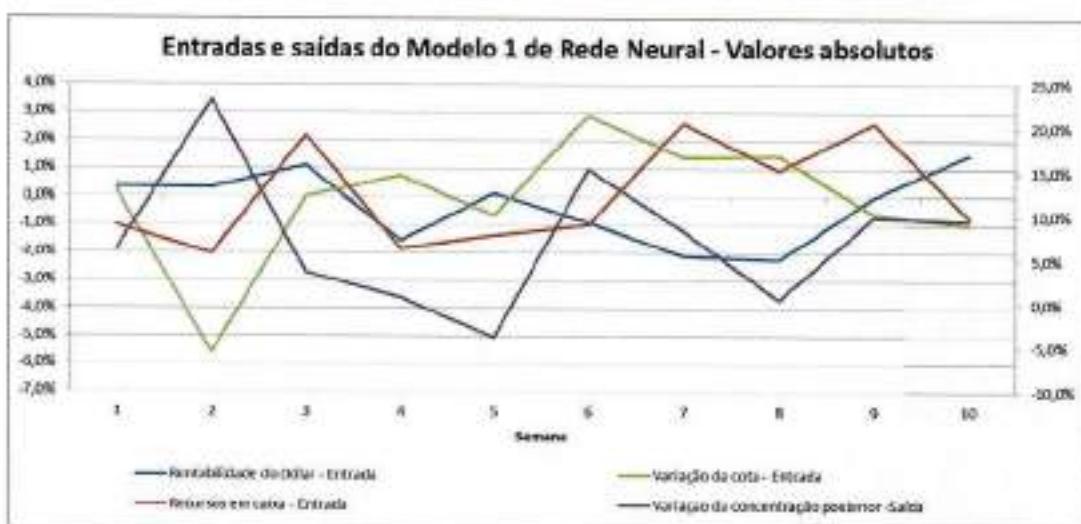
Figura 15 - Modelo 1 da rede neural - Fundo fictício



Fonte: Autor

Podemos visualizar a não linearidade do problema a partir do gráfico abaixo, que contém as entradas e saídas da rede neural. Os valores ilustrados referem-se a uma classe específica, o Dólar. É importante ressaltar que nesse modelo desenvolve-se uma rede para cada classe de ativo.

Figura 16- Entradas e saídas do Modelo 1 de Rede Neural - Valores absolutos

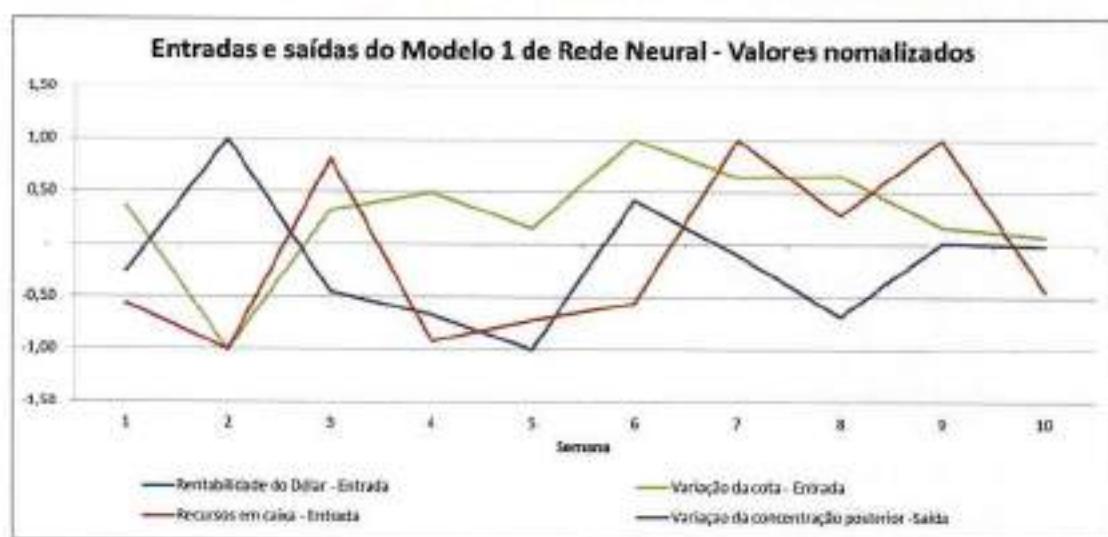


Fonte: Autor

Para melhor visualização do gráfico acima, as variáveis Recursos em caixa e Variação da concentração posterior são mensuradas pela escala secundária.

Abaixo temos o mesmo Modelo 1, mas agora com valores normalizados.

Figura 17- Entradas e saídas do Modelo 1 de Rede Neural - Valores normalizados



Fonte: Autor

4.4.2 Modelo 2 de rede neural – Fundo Fictício

Adicionalmente, foi criado um Modelo 2 para solução de nosso problema. Há três diferenças em relação ao modelo 1,

1. Leva-se em consideração a rentabilidade dos outros ativos
2. Na entrada há também a concentração na semana n-1;
3. A resposta fornecida contém a concentração por classe de ativo, ao invés de variação da concentração por classe de ativo.

Desta forma as entradas do Modelo 2 são:

- Rentabilidades das onze classes de ativos,
- Concentração das dez classes de ativos,
- Rentabilidade da cota.

E a saída é:

- Concentrações das dez classes de ativos.

Cabe destacar que no Modelo 2, temos apenas uma rede neural para criar, enquanto que no modelo 1, há uma rede neural para cada classe de ativo. Além disso as saídas da rede neural do Modelo 2 já são as concentrações em si, enquanto que na rede neural do Modelo 1 as saídas da rede neural são as variações de concentração.

4.5 Conclusões sobre o fundo fictício

Percebe-se que as relações entre entrada e saída são altamente não lineares e não sabemos ao certo como é a relação entre as variáveis.

Dada esta situação, e que desejamos utilizar redes neurais para construir o modelo, uma série de possibilidades aparecem como: Quais entradas serão utilizadas? Se forem utilizadas muitas variáveis de entrada, será que as entradas de

baixa correlação com a saída não irão prejudicar a previsão, ou o benefício de termos mais dados compensa?

Para responder às perguntas propostas foi necessário realizar uma grande quantidade de testes, pois há muitos parâmetros que podem ser alterados no treinamento e na construção da rede como: Taxa de aprendizado, número de nós, número de iterações, função de transferência, entre outros. Além disso, foi necessário testar diferentes entradas, com diferentes formas de pré-processamento dos dados.

Quanto a questões de entrada e saída, optamos por trabalhar com o Modelo 1 para treinamento da rede do fundo fictício Simula, pois como será visto na seção de Validação, o Modelo 2 apresenta um erro inaceitável, fato causado possivelmente pela dependência muito baixa entre as certas variáveis de entrada e de saída.

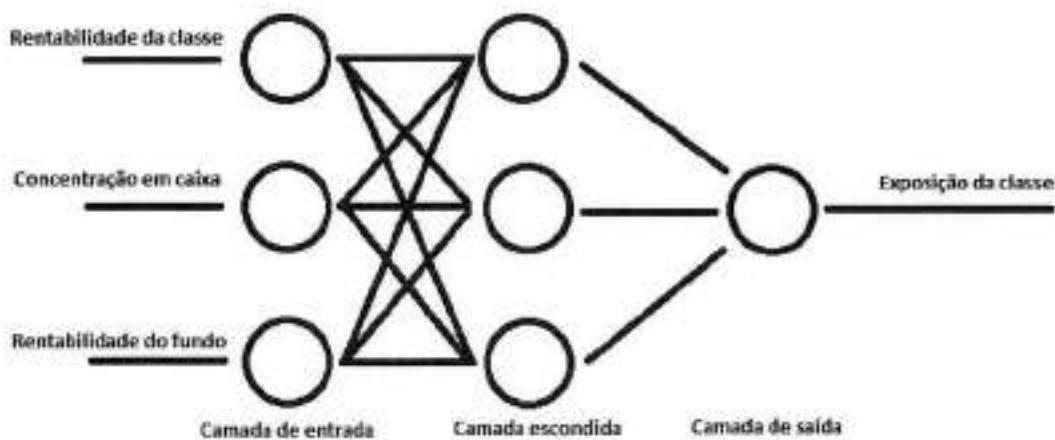
5 REDES NEURAIS TREINADAS PARA FUNDOS REAIS

Para a simulação inicial com dados de fundos reais, foi utilizado o modelo 1 de rede neural do fundo fictício, com a diferença da periodicidade dos dados ter sido alterada de uma semana para um mês. Posteriormente fizemos alterações convenientes na estrutura da rede, como a retirada da concentração em caixa como entrada para os fundos que realizam alavancagem, pois conforme comentado anteriormente essa variável não é relevante para esse tipo de fundo.

5.1 Modelos de rede neural utilizados

Temos nas figuras abaixo as quatro redes neurais que utilizamos para modelar os fundos reais iremos nomeá-las de A a D:

Figura 18- Rede neural A

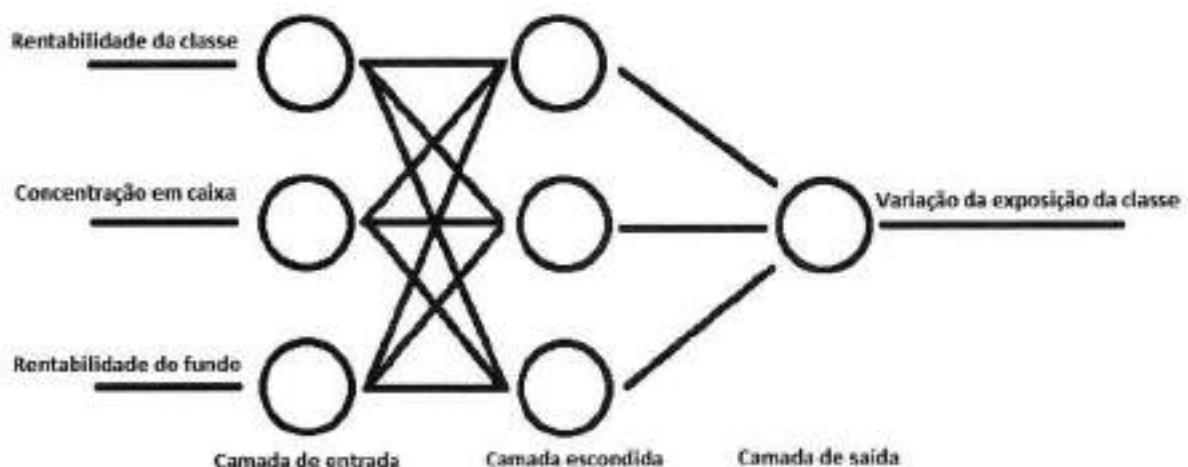


Fonte: Autor

Para a Rede Neural A, utiliza-se como entrada a rentabilidade da classe, a concentração em caixa e a Rentabilidade do fundo, e como saída, a exposição na

classe. Adicionalmente existe a possibilidade de normalizarmos qualquer uma dessas variáveis.

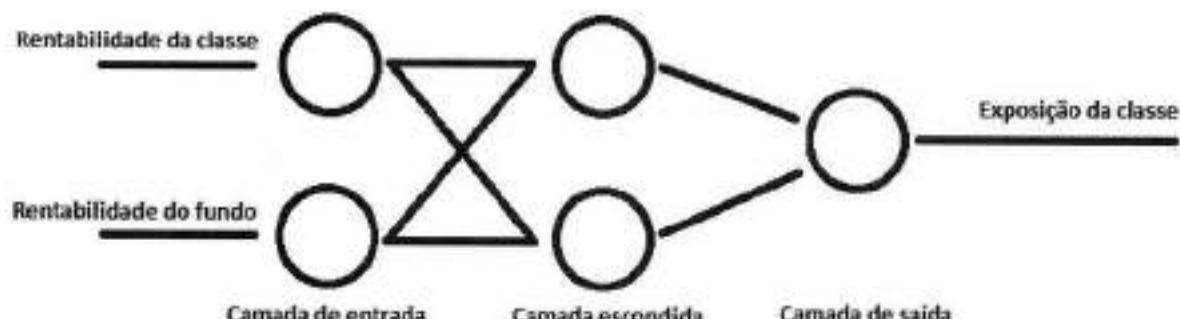
Figura 19- Rede Neural B



Fonte: Autor

Para a Rede Neural B, utiliza-se como entrada a rentabilidade da classe, a concentração em caixa e a Rentabilidade do fundo, assim como na Rede Neural A, porém como saída, temos a variação da exposição na classe. Assim como na Rede Neural A, existe a possibilidade de normalizarmos qualquer uma dessas variáveis.

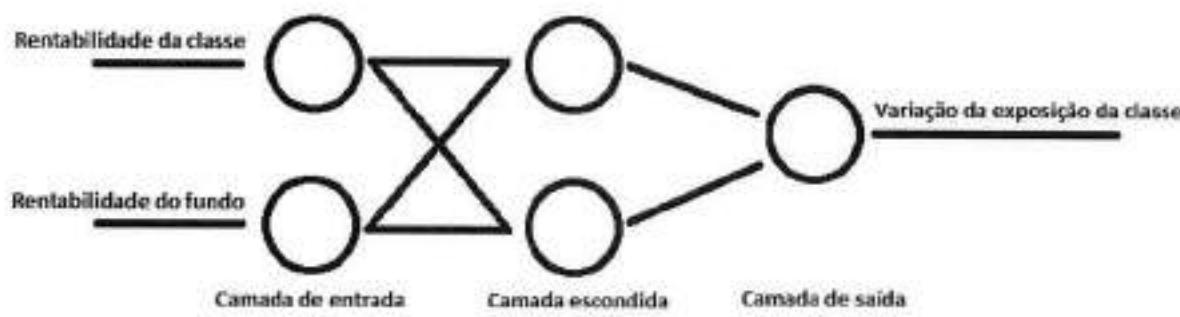
Figura 20- Rede neural C



Frontier Author

Para a Rede Neural C, utiliza-se como entrada a rentabilidade da classe e a rentabilidade do fundo, diferentemente das redes anteriores, esta não leva em consideração a concentração em caixa. Como saída, temos a exposição na classe. Assim como nas redes anteriores, existe a possibilidade de normalizarmos qualquer uma dessas variáveis. Destacamos que este tipo de rede será útil para os fundos que realizam alavancagem.

Figura 21- Rede neural D



Fonte: Autor

Para a Rede Neural D, utiliza-se como entrada a rentabilidade da classe e a rentabilidade do fundo. Como saída, temos a variação da exposição na classe. Assim como nas redes anteriores, existe a possibilidade de normalizarmos qualquer

uma dessas variáveis. Destacamos que este tipo de rede será útil para os fundos que realizam alavancagem.

5.2 Ferramentas de auxílio de escolha de rede neural

Com o objetivo de desenvolver um modelamento eficiente, é dado ao usuário do software interno do FEPREM a opção de normalizar os dados de entrada e saída da rede neural. Ressaltamos que os procedimentos descritos nesse item são realizados internamente e somente os resultados são disponibilizados ao cliente.

Além desse tratamento de dados, temos a opção de decidir se a concentração em caixa será entrada da rede neural. Com isso alteramos o modelo 1 adotado para o fundo fictício. Tal alteração se mostrou extremamente conveniente para os fundos em que o gestor possui a carteira alavancada, nesses casos recursos em caixa não é um fator limitante para aplicação em um ativo, e consequentemente essa variável prejudica o modelamento, conforme observamos em alguns testes. Deixamos como opção também a possibilidade de escolher como saída da rede neural a concentração por classe de ativo ao invés de variação da concentração por classe de ativo, utiliza-se essa opção nos casos em que a série temporal de variação de exposição é muito agressiva nas transições, o que dificulta a modelagem. Temos a possibilidade de implementar quatro tipos de rede neural.

Podemos também variar o passo, número de iterações, números de dados para treinamento, validação e teste.

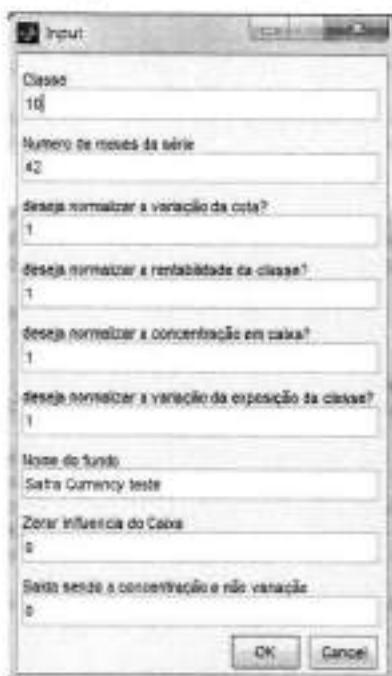
5.3 Algoritmos para automatização do processo de treinamento

Dado que foi necessário criar uma rede neural para cada classe de ativo, para cada fundo, temos um grande número de redes neurais – setenta –, já que modelamos 7 fundos com 10 classes cada. Esse fato, fez com que fosse necessário automatizar os seguintes processos para tornar o projeto viável:

- 1 Tratamento dos dados de entrada e saída da rede neural
- 2 Escolha de um dos quatro modelos de rede neural adotados (através da variação dos dois últimos campos da tela abaixo).
- 3 Exportação de todos os resultados obtidos, como gráficos, medidas de erro, valores previstos pela rede neural e pelo método linear.

Referente ao processo 1 e 2 automatizados: Conforme podemos ver na tela abaixo, é possível escolher a classe de ativo do fundo que queremos treinar, informar o números de meses que temos dados, a possibilidade de normalização dos dados de entrada e saída, o nome do fundo e por último, os dois parâmetros responsáveis por determinar qual dos quatro modelos iremos utilizar.

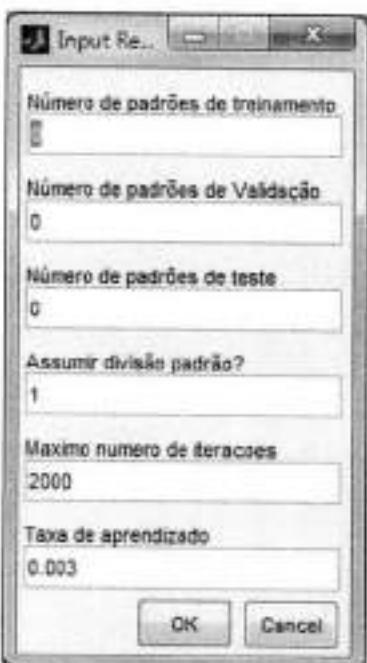
Figura 22- Tela 1 de inputs



Fonte: Autor

Abaixo vemos a segunda tela desenvolvida para automatização do processo.

Figura 23- Tela 2 de inputs

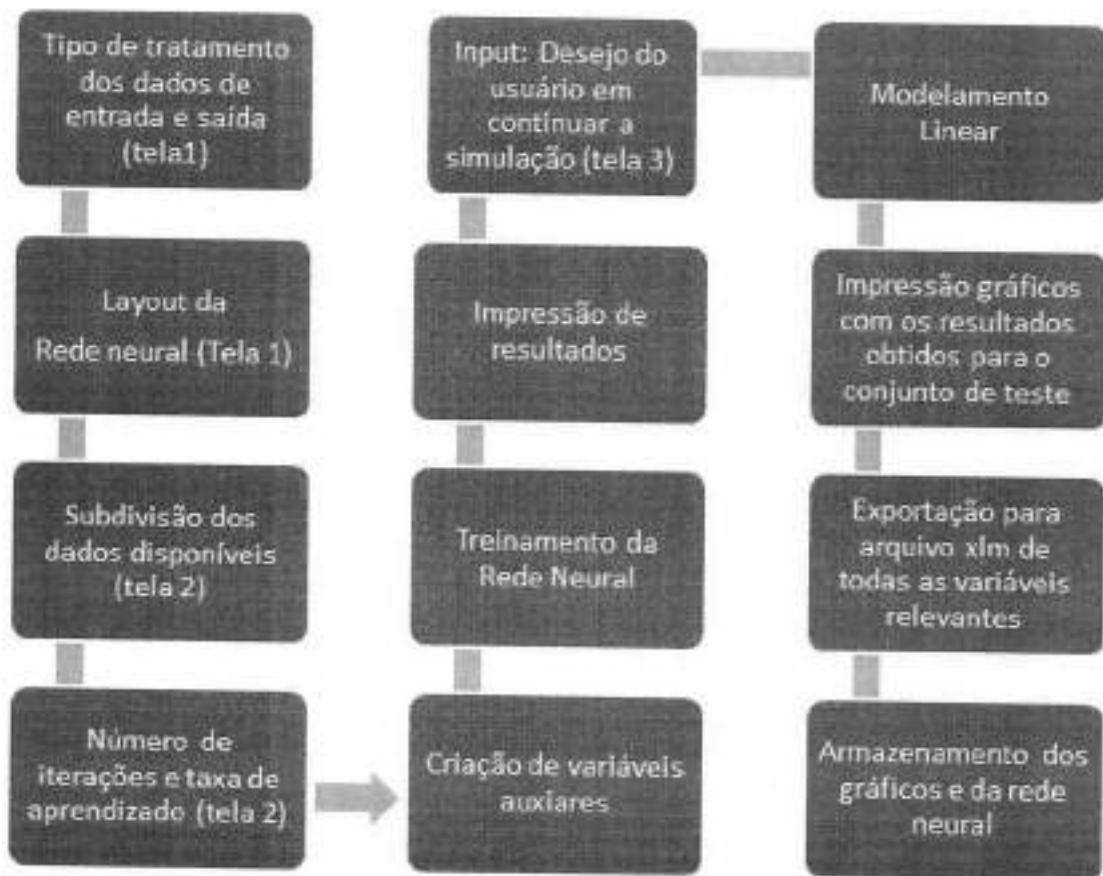


Fonte: Autor

Referente a segunda tela, os quatro primeiros campos referem-se a divisão dos dados disponíveis em subconjuntos de treinamento validação e teste, pode-se estabelecer uma divisão arbitrária ou então um divisão pré-estabelecida. Os dois últimos campos referem-se número de iterações e taxa de aprendizado. De modo geral quando o fundo apresentava erro no treinamento, diminuímos a taxa de aprendizado, e se com a alteração a rede passasse a funcionar, concluímos que provavelmente tratava-se de oscilação em torno do ponto de mínimo gradiente ou divergência.

Todos os parâmetros acima informados são então passados para o script criado, que segue um fluxo resumido abaixo em diagrama de blocos.

Figura 24- Fluxo no Matlab



Fonte: Autor

Os blocos um a quatro já foram explicados anteriormente (inputs do sistema), vamos agora detalhar a funcionalidade dos outros. O bloco cinco representa a etapa de consolidação dos inputs do usuários na definição da rede e separação dos dados em subgrupos. O bloco seis explicita a fase do treinamento da rede neural, na etapa seguinte o usuário vê os resultados obtidos para a rede e decide se deseja prosseguir, com isso evita-se perder tempo com processamento do programa caso os resultados obtidos com a rede neural sejam insatisfatórios.

O nono bloco no mostra a modelagem linear, utilizada para evidenciar a necessidade de redes neurais. Já os três últimos blocos tratam da impressão de resultados e exportação de informações e gravação da rede neural treinada.

5.4 Medidas de erro

Como forma de validar os resultados obtidos no nosso projeto, calcula-se o erro quadrático médio em dois sistemas.

Rede Neural – O cálculo de erro é feito com base nos dados de treinamento validação, e teste. Por conter uma maior conjunto de valores, o EQM do conjunto de treinamento será o mais decisivo na para verificação da eficiência da rede neural.

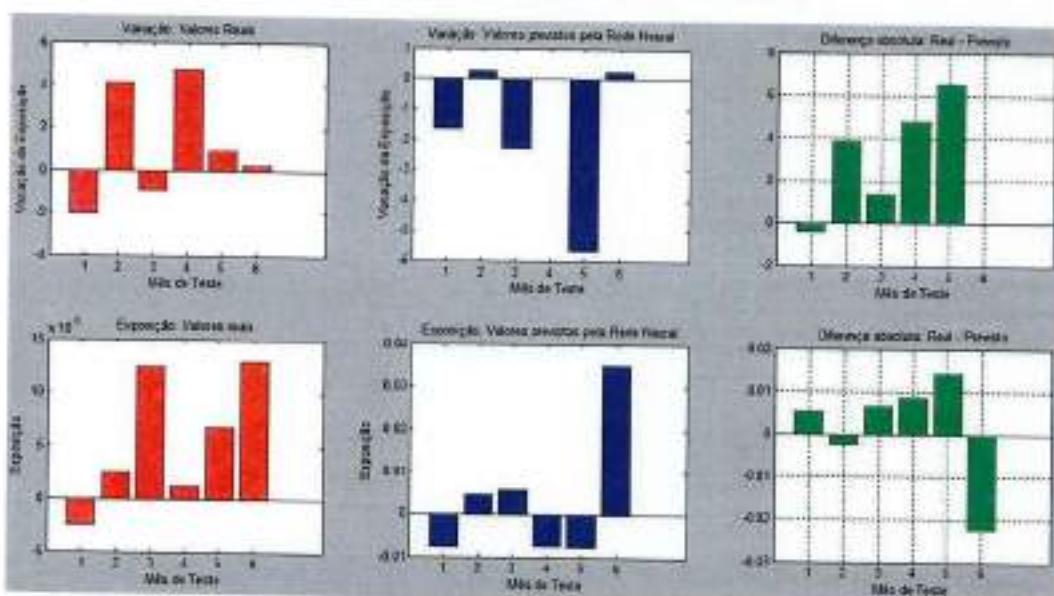
Sistema Final – Neste caso, calculamos o erro quadrático médio das exposições em si, ou seja, do dado final que é de interesse do cliente. Este erro é igual ao obtido pela rede neural nos casos em que escolhemos como saída da rede neural a concentração (não normalizada), para todos os outros casos os erro diferem.

$$\text{Erro Quadrático médio} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (1)$$

5.5 Validação

Como forma de validar inicialmente a eficiência da rede neural, temos a seguinte tela. Ela é resultado dos parâmetros que informamos anteriormente nas telas de input.

Figura 25 - Output dos resultados da Rede Neural



Fonte: Autor

Os três primeiros gráficos (parte de cima da tela) referem-se a saída da rede neural, no caso *variação de exposição* como pode verificar pelo nome do eixo dos gráficos. Temos respectivamente os valores reais, os calculados pela rede e a diferença absoluta entre eles. Na parte inferior da tela, temos respectivamente: os valores reais de concentração, os calculados pela rede e diferença absoluta entre eles. No caso projeto, é essa informação que iremos efetivamente fornecer ao cliente, ou seja a concentração ou exposição por classe de ativo. Após a visualização desse gráfico, podemos optar pelo prosseguimento do programa, cálculo com método linear e gravação de todos os resultados, redes e gráficos, alternativamente podemos optar pela interrupção do programa. A opção de interrupção é interessante quando estamos procurando por uma rede mais eficiente e ocasionalmente escolhemos parâmetros que geram redes piores, que não desejamos gravar resultados.

Uma vez observado que o último gráfico possui valores razoáveis de erro absoluto, grava-se todos o resultados e a rede treinada. No desenvolvimento do projeto, fizemos diversas alterações nas redes neurais de cada classe de ativo, obtendo muitas vezes bons resultados, mas há casos em que não foi possível chegar a resultados razoáveis. Na apresentação de resultados no site é mostrado todos os erros obtidos (o da Rede Neural e o do Sistema), dessa forma o cliente

(gestor de patrimônio) tem uma medida de confiabilidade dos resultados daquela classe. Também na apresentação de resultados temos a modelagem com método linear.

5.5.1 Método linear

Com o objetivo de evidenciar a superioridade da modelagem com rede neural para nosso problema, desenvolvemos uma rede neural com um nó e função de transferência linear, referenciado nesse trabalho como *Método Linear*. Para todo cálculo que efetuamos com a rede neural, calculamos também com o método linear e comparamos os erros e os gráficos. Essas informações são disponibilizadas ao cliente, não apenas para efeito de comparação, mas para uso da informação também.

Há alguns poucos casos em que realmente o uso do método linear é mais eficiente e válido, para estes casos o cliente poderá consultar esses resultados em vez dos calculados pela rede neural. Dessa forma atendemos o objetivo do gestor, mesmo que não pelo método proposto originalmente.

6 CRIAÇÃO DO SITE E BANCO DE DADOS

6.1 Domínio e servidor de hospedagem

O primeiro passo para o desenvolvimento de um aplicativo que será acessado via web é garantir a existência de um computador onde seus arquivos serão armazenados - o servidor de hospedagem - e também um endereço que apontará para este computador - o domínio.

Dentre os fatores relevantes para a tomada de decisão relacionada à escolha de um servidor de hospedagem, podemos destacar os seguintes:

- Distância física entre computador utilizado como servidor de hospedagem e computador do usuário que acessa o aplicativo. Esta distância pode afetar o tempo necessário para acessar o aplicativo, através de uma medida denominada *ping*, que mede o tempo que um determinada requisição demora para atingir o servidor somado ao tempo de chegada da resposta a esta requisição.
- Confiabilidade do servidor: determinada pelo tempo em que o servidor permanece online ininterruptamente. Inversamente, pode-se pensar que um melhor servidor seria aquele em que quase não há quedas de conexão.
- Suporte técnico da empresa responsável: mesmo que o servidor esteja quase sempre online e esteja a uma distância física relativamente próxima do usuário, pode haver ainda outros problemas que prejudiquem o bom funcionamento de uma aplicação hospedada. Deve-se nesse caso contar com a assistência da empresa responsável pelo servidor, que pode acessar e configurar seu software/hardware com facilidade.

Quanto ao domínio a ser escolhido, podemos nomear o seguinte fator como importante para sua escolha:

- Facilidade de memorização e escrita: no caso de uma aplicação web comercial principalmente, já que o usuário deve considerar o nome simples, elegante e de fácil reprodução. Domínios longos ou confusos podem ser um impedimento para que o usuário acesse o aplicativo.

6.1.1 Framework utilizado

O processo de criação de um aplicativo web é extenso. Conforme o projeto se desenvolve, pode vir a se tornar desorganizado e inconsistente. Além disso, diversas fases desse processo de criação são muito parecidos para os mais diversos tipos de aplicações. Então, para que não haja repetição de trabalho e pensando em torná-lo consistente e livre de erros, utilizaremos em nosso projeto um *framework*.

Frameworks são uma combinação de bibliotecas de programação, conceitos de arquitetura de software e convenções que se estabelecem, que têm como objetivo tornar o desenvolvimento de software mais ágil e livre de erros.

Para o caso da aplicação FEPREM, como a linguagem de programação utilizada é o PHP, devemos escolher um *framework* específico para esta linguagem. Existem diversas opções disponíveis. Contudo, por familiaridade e facilidade de uso, utilizaremos o *framework* CakePHP. Neste *framework*, basta seguir suas convenções de nomenclatura e organização de diretórios/funções/variáveis para que a arquitetura MVC seja implementada sem maiores problemas.

6.1.2 Detalhes da camada Model

Vamos agora detalhar a camada Model de nosso aplicativo. Inicialmente será composta por três modelos principais: Fundo, Cota e Distribuição.

6.1.2.1 Model: Fundo

O modelo **Fundo** representa, como fica óbvio pelo nome, os fundos cadastrados em nosso aplicativo. Conterá apenas informações gerais relacionadas ao fundo e que sejam pertinentes à nossa aplicação, como por exemplo o nome do fundo, seu gestor ou patrimônio líquido.

6.1.2.2 Model: Cota

O modelo **Cota** representa uma única cota , pertencente a um determinado fundo em um determinado dia. Portanto, contém apenas o fundo a que pertence, o valor da cota e a data em que foi registrada.

6.1.2.3 Model: Distribuição

O modelo **Distribuição** representa a porcentagem que um determinado fundo de investimento investe num dado setor em um, para uma determinada data. Por consequência deve conter estas informações: fundo, tipo de investimento, data e porcentagem. O conjunto de porcentagens quando fixamos o fundo e a data é o que representa a distribuição.

6.1.2.4 Relação entre os modelos

Tão importante quanto descrever o que cada modelo representa no mundo real individualmente é explicitar a relação que entre eles. Para que tal relação possa ser explicada de maneira mais fácil, observemos abaixo. Nela fica clara a relação de dependência que deve haver entre o fundo, suas cotas e sua distribuição.

Figura 26 - Esquema da Camada Model



Fonte: Autor

Um único fundo deve possuir várias cotas (uma para cada dia). Isso está explicitado na figura acima pela reta que liga os círculos de cada modelo e pela representações do algarismo 1 próximo ao modelo Fundo e de um asterisco próximo ao modelo Cota. A mesma informação vale para a relação existente entre os modelos Fundo e Distribuição. Um único fundo deve possuir várias distribuições, uma para cada tipo de investimento e data.

Inversamente, e ainda observando o modelo da figura acima, podemos pensar que uma cota ou distribuição deve pertencer a um único fundo, nunca havendo por exemplo uma cota que pertença a dois fundos diferentes. O próprio modelo já cuida para que isso não ocorra e este tipo de garantia de consistência é de responsabilidade do próprio *framework*.

6.1.3 Ferramentas de interface

A interface do aplicativo, que será exibida pelo navegador na tela do usuário, necessita de uma outra gama de ferramentas, linguagens e frameworks que serão detalhados abaixo. Sua implementação é tão importante quanto à implementação das camadas Model e Controller, visto que a aparência de um aplicativo pode ser o detalhe que afasta ou atrai o usuário.

Em nosso aplicativo decidimos deixar a fase de aperfeiçoamento da interface para um estágio final de desenvolvimento. Isto porque inicialmente devemos garantir a funcionalidade de nossa aplicação, considerando primeiramente a rede neural e também o fluxo dos dados na aplicação, sendo resgatados do bando de dados para a tela do usuário. Deve-se garantir inicialmente uma interface básica para testes e então aperfeiçoá-la conforme o projeto se aproxima do fim.

6.1.4 Interface básica – Bootstrap

Para cuidar dos elementos básicos de interface – i.e. – que estarão dispostos em praticamente todas as telas do programa, utilizaremos *um framework frontend*

determinado *bootstrap*. Trata-se de uma ferramenta que pode ser usada livremente, sem nenhum tipo de licença especial.

Esta ferramenta já contém diversos elementos importantes para a determinação da interface de um aplicativo, todos pré-formatados de maneira elegante e facilmente adaptáveis para qualquer tamanho de tela. Alguns destes elementos foram retirados da página oficial do *framework* e colocados na figura abaixo, como prévia de como será a interface do aplicativo ao utilizar tais elementos.

Figura 27 - Elementos básicos do framework Bootstrap



Fonte: Autor

Além de elementos gráficos pré-programados, o *bootstrap* conta ainda com diversas funcionalidades importantes quando se trata de aplicações web e que são também comuns a muitas. Como exemplos importantes, podemos citar barras de progresso, navegação por abas, menus *dropdown* e *slideshows*.

6.1.4.1 Ferramenta de exibição de gráficos - Google Charts

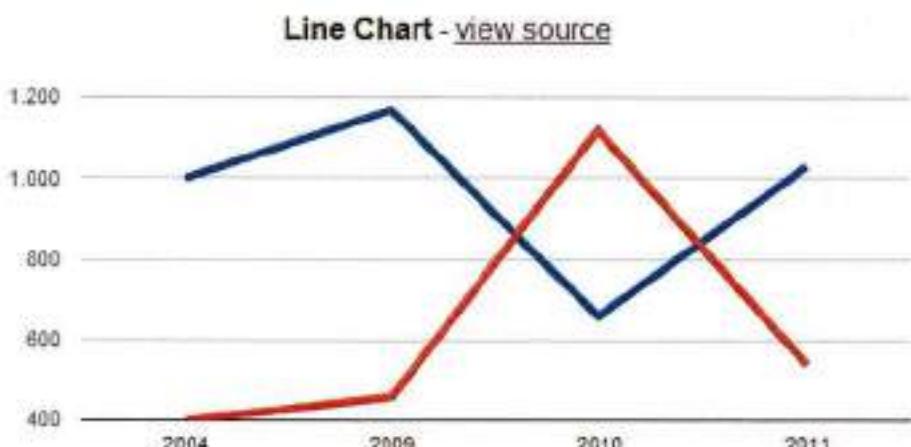
Por se tratar de uma aplicação voltada para o mercado financeiro, é importante em nosso caso possuir uma ferramenta de exibição de gráficos que tenha boa qualidade e velocidade de carregamento rápida.

Após considerar diversas opções de bibliotecas de exibição de gráficos, chegamos ao que elegemos a melhor para nossas necessidades, o *Google Charts*. Trata-se de uma ferramenta de exibição de gráficos simples, de implementação extremamente simples. Além disso, tem extensa documentação, que é constantemente atualizada pela equipe Google.

Para a passagem dos diversos dados que irão montar o gráfico, utiliza-se uma notação que se chama JSON. Trata-se de um conhecido formato para troca de dados computacionais multi-linguagem. É uma alternativa ao conhecido XML.

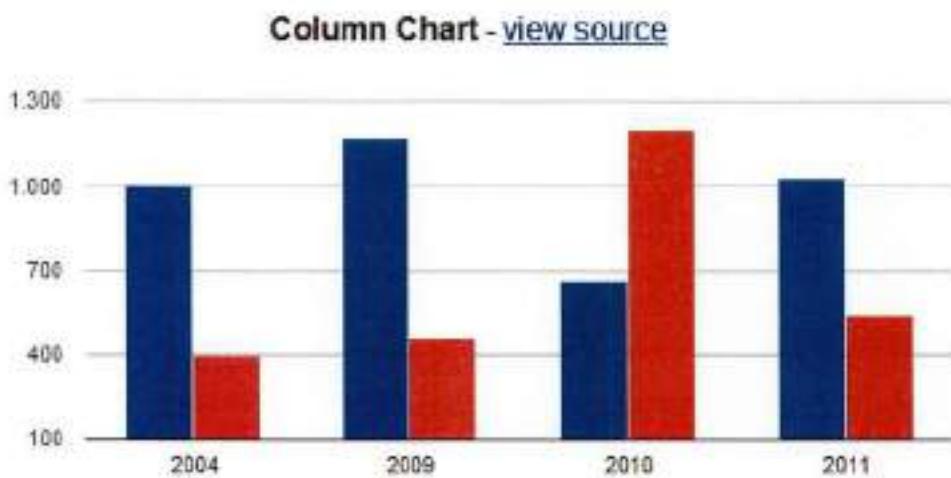
A aparência final dos gráficos gerados pela ferramenta Google Charts pode ser conferida nas figuras abaixo. Estas foram retiradas diretamente do site oficial do Google Charts.

Figura 28- Exemplo de gráfico de linha gerado pela ferramenta Google Charts



Fonte: developers.google.com

Figura 29- Exemplo de gráfico de barras gerado pela ferramenta Google Charts



Fonte: developers.google.com

6.2 Aspectos operacionais

Após a finalização dos modelos e o término do site é necessário definir como será feita a rotina operacional de manutenção do site. Como o produto apresenta dados dos fundos (como cotas), que são informadas diariamente, e

estimativas da concentração por classe de ativo, que serão informadas mensalmente, é necessário definir como serão feitas as atualizações, de modo a prover confiabilidade aos dados e agilidade.

Diariamente, o banco de dados deverá ser alimentado com as rentabilidades de cada classe de ativos, assim como o valor da cota de cada fundo. O valor da cota de cada fundo e os valores de concentração por classe de ativo do fundo (com seis meses de atraso) podem ser obtidos no site da CVM (Comissão de Valores Mobiliários - www.cvm.gov.br). Já as rentabilidades dos índices setoriais da bolsa de valores podem ser obtidos pelo site da BM&F Bovespa (www.bmfbovespa.com.br), e as rentabilidades relativas a renda fixa podem ser calculadas pelos dados divulgados pela ANBIMA (Associação Brasileira das Entidades dos Mercados Financeiro e de Capitais – www.anbima.com.br), sendo o IMA-B5 calculado a partir de uma média dos preço de mercado dos títulos públicos NTN-B com vencimento menor que 5 anos, e o IMA-B5+ para NTN-B com vencimento maior que 5 anos. A cotação do CDI pode ser obtido no site da CETIP (www.cetip.com.br) e finalmente, a cotação do Dólar é obtida no site do Banco Central (www.bcb.gov.br).

Todos os dados descritos acima são de domínio público, no entanto a obtenção diária é trabalhosa, pois no site da CVM, por exemplo, é necessário acessar a página de um fundo por vez, tornando o processo extremamente ineficiente.

Uma alternativa ao processo manual é a compra dos dados da ANBIMA. Neste caso todos os dados são disponibilizados em um mesmo arquivo, de fácil automatização para inclusão no banco de dados. O valor dos dados é de aproximadamente R\$1.500 semestrais.

Para a fase não comercial do projeto, a obtenção dos dados é feita manualmente, porém em um segundo momento, quando o produto for comercializado, os dados da ANBIMA poderão ser comprados para automatizar a inclusão diária de dados.

Além dos dados diários, mensalmente faremos estimativas das concentrações por classe de ativo, e para isto, os dados do mês são inseridos no modelo do Matlab e as respostas são exportadas e inseridas no banco de dados.

Uma vez por mês, as redes receberão um novo treinamento, incluindo os dados oficiais de concentração por classe de ativo do fundo divulgados pela CVM (com um atraso de 6 meses).

7 RESULTADOS

7.1 Estratégia de testes

Para a realização dos testes, decidimos utilizar a ferramenta do Matlab para redes neurais. A escolha do Matlab foi motivada pela confiabilidade, pois esse toolbox é largamente utilizado por pesquisadores em todo mundo. Além disso, a ferramenta é altamente automatizada e fornece medidas de análise de resultados.

A estratégia do grupo foi construir a rede a partir dos dados do fundo fictício Simula e encontrar os parâmetros ideais de treinamento como: o tamanho da rede, o passo de aprendizagem, número de iterações entre outros fatores. Assumimos então que para prever o fundo real já tínhamos parâmetros próximos do ideal, e fizemos apenas pequenos ajustes para os novos treinamentos.

Para viabilizar a expressiva quantidade de testes necessários, foi preciso criar um programa em Matlab que automatizasse a aquisição e o tratamento dos dados e mostrasse os resultados de maneira objetiva e concisa, com diversos gráficos, os quais veremos a seguir nos resultados de ensaios.

Com o objetivo de desenvolvermos uma modelagem eficiente, implementamos diversas redes neurais com base nos Modelos 1 e 2, que foram detalhados anteriormente. Variamos os seguintes parâmetros:

1. Função de transferência dos nós;
2. Método de treinamento;
3. Número de iterações;
4. Taxa de aprendizado;

Quanto ao tratamento dos dados de entrada e saída da rede neural, trabalhamos por enquanto com:

- Normalização de zero a um;
- Valores absolutos ;
- Normalização de menos um e um.

Apresentaremos a seguir uma série de ensaios, destacando os parâmetros da rede neural e o tratamento dos dados de entrada e saída. Mostraremos também os erros obtidos em cada teste.

Antes de começarmos a apresentação dos ensaios, faremos um breve comentário sobre as variáveis da rede neural (explicação completa na seção Construção de Modelos).

Em alguns casos adotamos como saída da rede neural a concentração por classe de ativo (que é o objetivo final), já em outros casos foi adotado a variação da concentração da classe de ativo. Essas variáveis de saída são altamente dependentes de três fatores:

- Recursos disponíveis para aplicação (caixa);
- Sua rentabilidade;
- Rentabilidade da cota.

Essas informações se relacionam da seguinte forma: A concentração de uma ativo dentro de um fundo depende da rentabilidade dele na semana e da quantidade de recursos que estão parados, em caixa. Conforme detalhado na seção Algoritmo de Criação.

As variáveis de saída são também dependentes, porém em menor escala, dos seguintes números:

- Rentabilidades das outras classes de ativos;
- Concentração das demais classes de ativos.

7.2 Fundo fictício

Detalharemos abaixo os resultados obtidos em ensaios com o fundo fictício:

7.2.1 Ensaio 1

Esse era nosso modelo inicial de implementação da rede neural (Modelo 1), dada a alta intuitividade na construção de uma única caixa-preta responsável pela solução completa de nosso problema. Abaixo as definições das variáveis de entrada e saída da rede neural:

- Entradas: Rentabilidade de todas as classes de ativos, concentração de todas as classes de ativos na semana anterior e variação da cota do fundo.
- Saída: Concentrações de todas as classes de ativos na semana atual.

Adotou-se os seguintes parâmetros para a Rede:

- Função de transferência dos nós: Tangente hiperbólica
- Método de treinamento: *Gradient descent backpropagation*
- Número de iterações: 5000
- Taxa de aprendizado: 0.01
- Entradas e saídas não normalizadas.
- Números de nós escondidos: 3
- Número de padrões de treinamento: 220
- Número de padrões de validação: 20
- Número de padrões de teste: 12

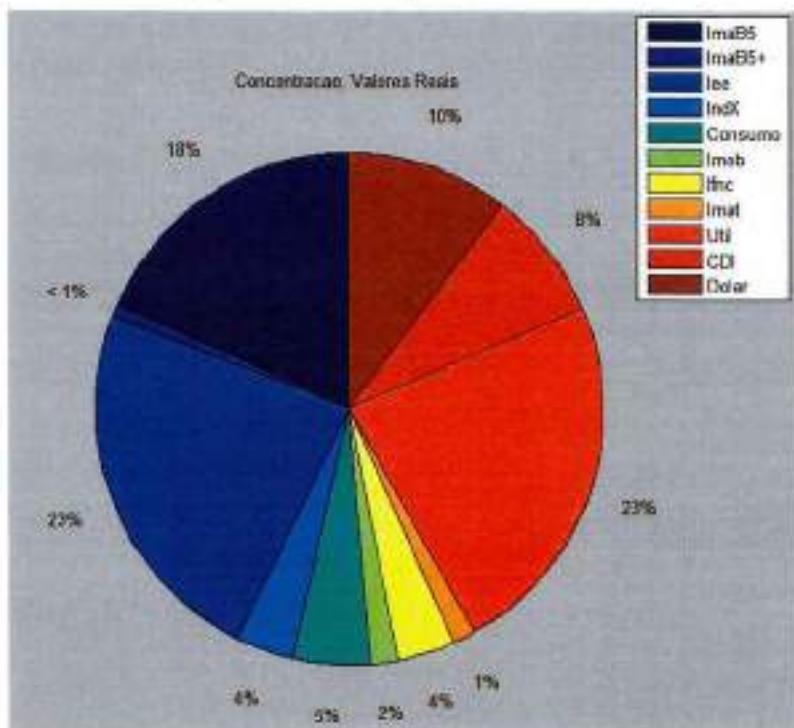
7.2.1.1 Resultados do Ensaio 1

Conforme podemos visualizar nas figuras abaixo os resultados deram uma dimensão da concentração por classe de ativo, contudo os números obtidos foram insatisfatórios, dada as grandes disparidades nas concentrações das classes menos significativas.

Apenas para exemplificar: os índices concentração em Imab5 e IEE (Índice de energia elétrica da BM&F) ficaram próximos aos reais, 16% previsto contra 18%

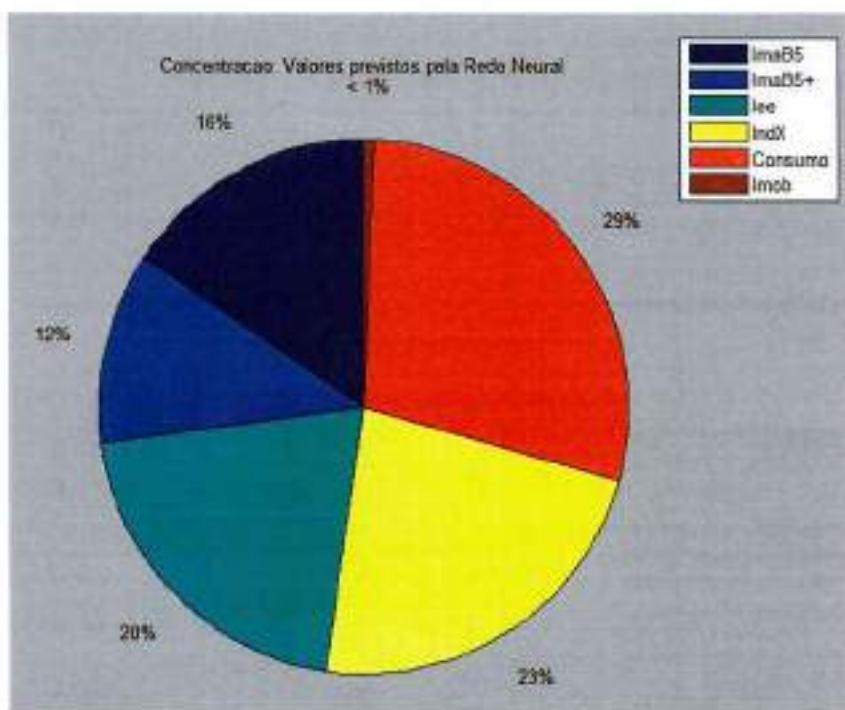
reais e 20% previsto contra 23% reais respectivamente, como podemos observar nos gráficos gerados pelo Matlab e reproduzidos abaixo.

Figura 30- Concentração do Fundo Ficticio



Fonte: Autor

Figura 31 - Concentração Prevista do Fundo Fictício



Fonte: Autor

Medida de erro: Para valores não normalizados o EQM – erro quadrático médio dos testes foi de 33%.

7.2.2 Ensaio 2

Esse nosso modelo de rede neural foi implementado visando isolar as variáveis mais importantes das de importância secundária, abaixo as definições das variáveis de entrada e saída:

- Entradas:
 - Rentabilidade da classe de ativo, no caso Dólar
 - Concentração de recursos em caixa na semana anterior
 - Variação da cota do fundo.
- Saída:
 - Variação da Concentrações da classe de ativo.

Adotou-se os seguintes parâmetros para a Rede referente a classe de ativo Dólar:

- Função de transferência dos nós: Linear
- Método de treinamento: *Gradient descent backpropagation*
- Número de iterações: 8000
- Taxa de aprendizado: 0.001
- Entradas normalizadas de menos um à um e saída não normalizada.
- Números de nós escondidos: 3
- Número de padrões de treinamento: 220
- Número de padrões de validação: 20
- Número de padrões de teste: 12

7.2.2.1 Resultados do Ensaio 2

Deve-se ressaltar que as saídas da rede não são as concentrações em si, mas sim a variação delas, número que combinado com as concentrações anteriores, resulta na concentração atual, conforme a formula abaixo:

$$\text{Concentração}(n) = \text{Concentração}(n - 1) \cdot \text{Rentabilidade}(n) \quad (2)$$

O erro quadrático médio obtido foi:

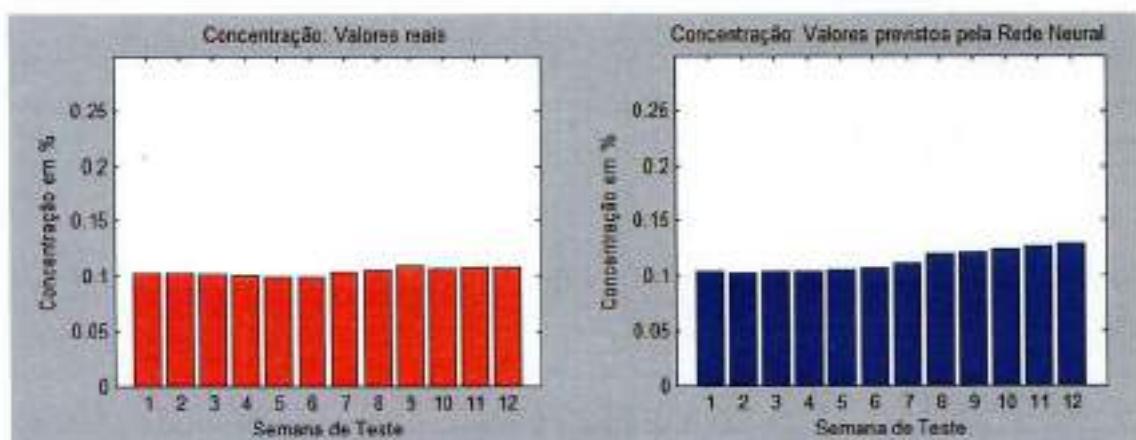
- 0.01243 para o conjunto de treinamento,
- 0.01384 para o conjunto de validação,
- 0.00166 para o conjunto de teste.

Esse baixo erro pode ser visualizado nos gráficos abaixo. Podemos dizer que essa rede neural é satisfatória para previsão da concentração em Dólar, porém como veremos no ensaio a seguir, essa mesma rede não funciona para outras

classes de ativos, esse fato nos motivou a adotar uma rede neural para cada classe de ativo.

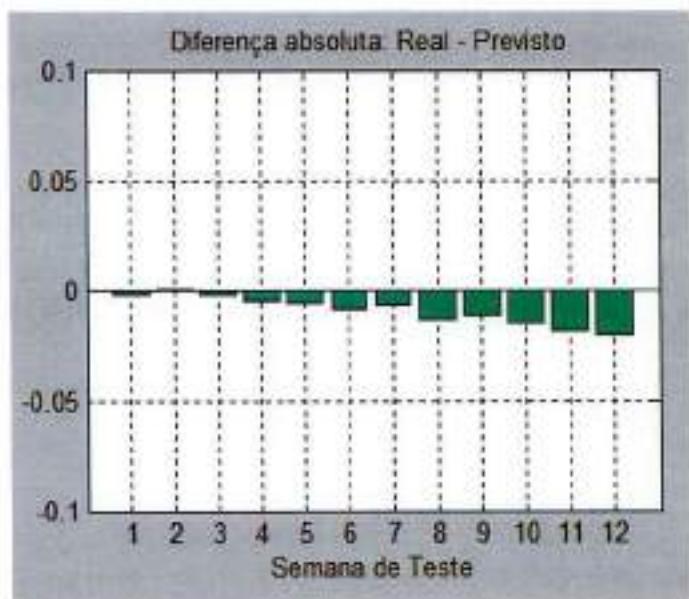
Dada a peculiaridade de cada variável, podemos mudar os parâmetros da rede neural de forma mais conveniente a obter erro mínimo.

Figura 32- Concentração de dólar no fundo Simula, valores reais e previstos



Fonte: Autor

Figura 33- Erro na previsão de concentração de dólar



Fonte: Autor

7.2.3 Ensaio 3

Nesse ensaio utilizamos as mesmas entradas e saídas do ensaio 2 e os mesmos parâmetros:

Adotou-se os seguintes parâmetros para a Rede referente a classe de ativo *IMAB5+*:

- Função de transferência dos nós: Linear
- Método de treinamento: *Gradient descent backpropagation*
- Número de iterações: 8000

- Taxa de aprendizado: 0.001
- Entradas normalizadas de menos um à um e saída não normalizada.
- Números de nós escondidos: 3
- Número de padrões de treinamento: 220
- Número de padrões de validação: 20
- Número de padrões de teste: 12

Neste ensaio diferentemente do anterior, tentamos prever a concentração da classe IMA-B5+ dentro do Fundo Simula.

7.2.3.1 Resultados do Ensaio 3

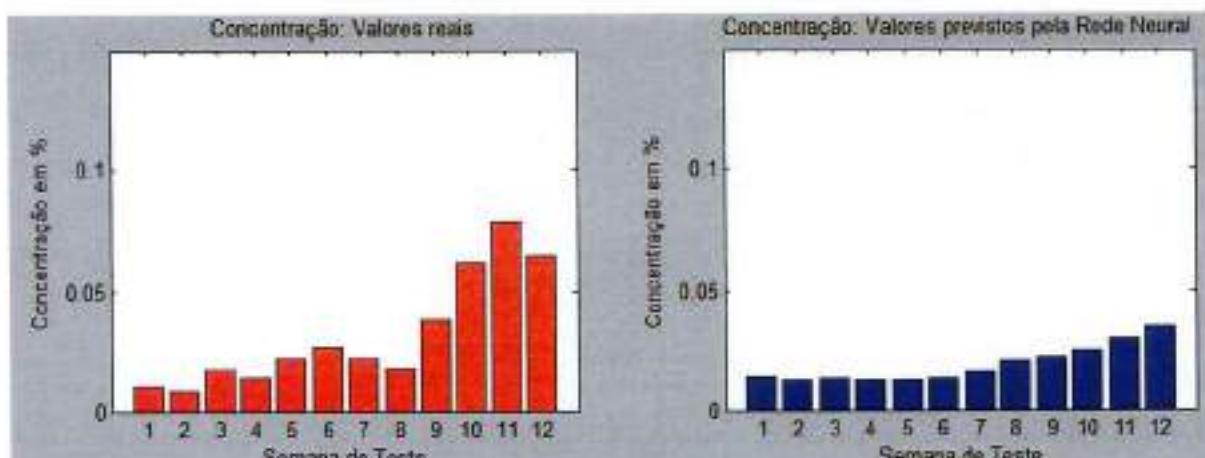
MSE para o conjunto de treinamento: 0.15948

MSE para o conjunto de validação: 0.18303

MSE para o conjunto de teste: 0.27218

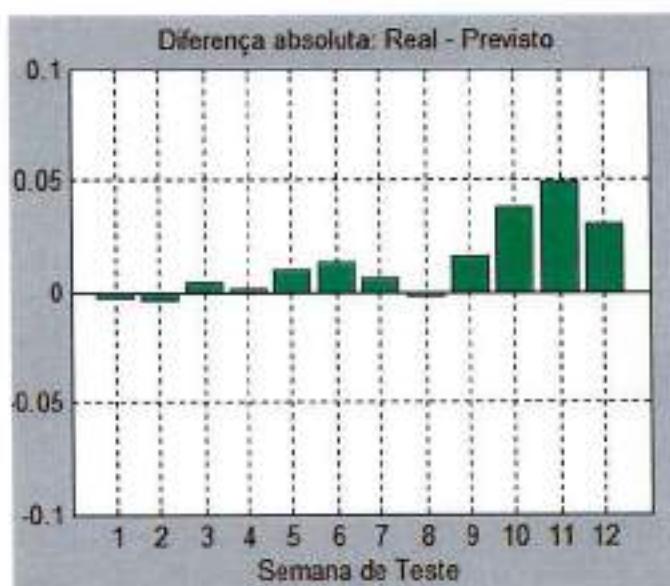
Como podemos ver nos gráficos abaixo, essa nossa rede neural previu a concentração de forma regular para as quatro primeiras semanas, mas daí em diante o erro aumento significativamente, o que mostra que a rede deve ser aprimorada.

Figura 34- Concentração do IMA-B5+ no Fundo Simula, valores reais e previstos



Fonte: Autor

Figura 35- Erro na previsão de concentração do IMA-B5+



Fonte: Autor

7.2.4 Ensaio 4

Para este ensaio, utilizou-se as seguintes entradas e saídas:

- Entradas:
 - Rentabilidade da classe de ativo, no caso IMAB-5
 - Concentração de recursos em caixa na semana anterior
 - Variação da cota do fundo.
- Saida:

- o Variação da Concentrações da classe IMAB-5.

Adotou-se os seguintes parâmetros para a Rede referente a *classe de ativo* IMAB-5:

- Função de transferência dos nós: Linear
- Método de treinamento: *Gradient descent backpropagation*
- Número de iterações: 5000
- Taxa de aprendizado: 0,009
- Entradas normalizadas de menos um à um e saída não normalizada.
- Números de nós escondidos: 3
- Número de padrões de treinamento: 220
- Número de padrões de validação: 20
- Número de padrões de teste: 12

Diferentemente, dos ensaios anteriores, neste diminuimos o número de iterações e aumentamos a taxa de aprendizado, obtivemos os seguintes erros quadráticos médios.

MSE para o conjunto de treinamento: 0.00675

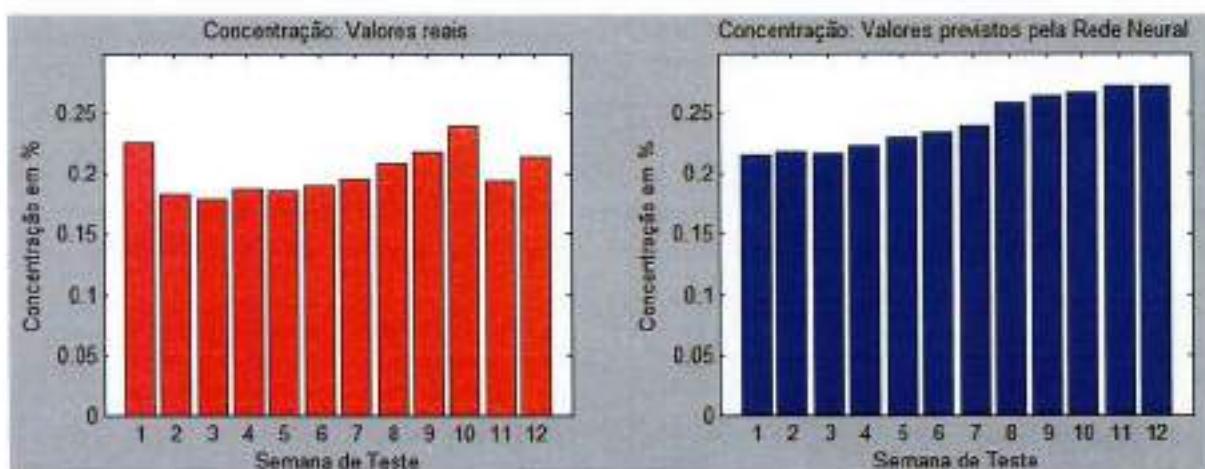
MSE para o conjunto de validação: 0.00691

MSE para o conjunto de teste: 0.01068

7.2.4.1 Resultados do Ensaio 4

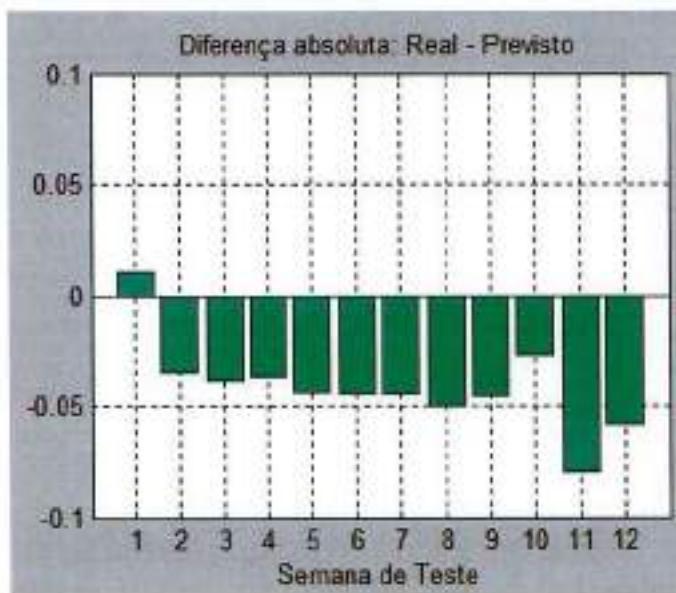
Como podemos concluir, os erros obtidos foram baixos e a rede neural está modelando bem a concentração do índice IMA-B5. Pode ser visto nos gráficos abaixo que o maior erro aconteceu no teste da semana 11, mas nos outros pontos não obtivemos erro maior que 5% na concentração.

Figura 36- Concentração do IMA-B5 no Fundo Simula, valores reais e previstos



Fonte: Autor

Figura 37 - Erro na previsão de concentração do IMA-B5



Fonte: Autor

7.2.5 Ensaio 5

Este ensaio visa encontrar uma rede neural que modele o comportamento da concentração da classe de ativo Industria.

Foram adotados diversos parâmetros, abaixo os que geraram os melhores resultados.

- Entradas:
 - Rentabilidade da classe de ativo, no caso Índice de Indústria
 - Concentração de recursos em caixa na semana anterior
 - Variação da cota do fundo.
- Saída:
 - Variação da Concentrações da classe Índice de Indústria.

Adotou-se os seguintes parâmetros para a Rede referente a *classe de ativo* Índice de Indústria:

- Função de transferência dos nós: Tangente hiperbólica
- Método de treinamento: *Gradient descent backpropagation*
- Número de iterações: 5000
- Taxa de aprendizado: 0.008
- Entradas normalizadas de menos um à um e saída não normalizada.
- Números de nós escondidos: 6
- Número de padrões de treinamento: 220
- Número de padrões de validação: 20
- Número de padrões de teste: 12

7.2.5.1 Resultados do Ensaio 5

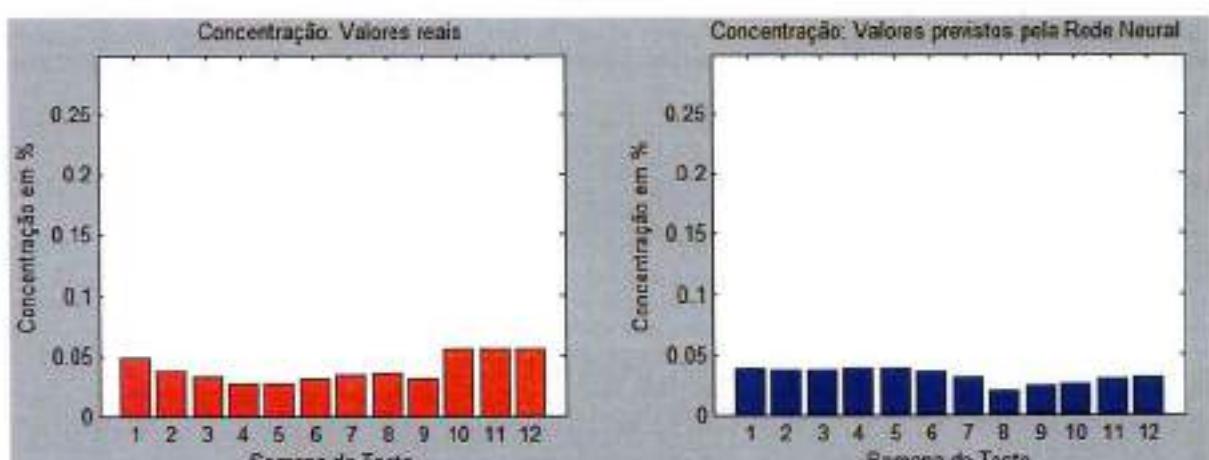
Abaixo os resultados de erro obtidos e os gráficos referentes aos testes realizados:

MSE para o conjunto de treinamento: 1.57705

MSE para o conjunto de validação: 0.02817

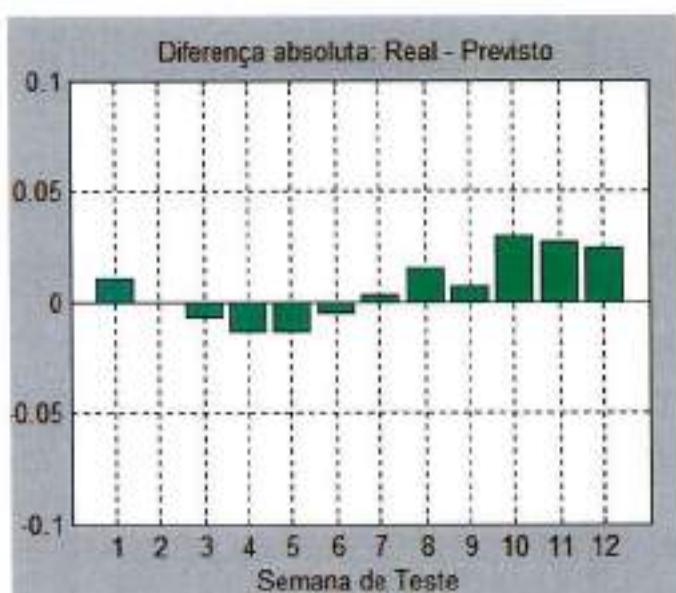
MSE para o conjunto de teste: 0.05929

Figura 38 - Concentração do Índice Indústria no Fundo Simula, valores reais e previstos



Fonte: Autor

Figura 39 - Erro na previsão de concentração do Índice de Indústria



Fonte: Autor

A primeira vista poderíamos ser tentados a concluir que a rede neural fez uma boa modelagem da variação de concentração do *Índice de Indústria*, já que os resultados obtidos foram bons. Isto contudo seria um grave erro, uma vez que o erro quadrático médio da série de treinamento foi alto: mais de 1.57. Como o conjunto de treinamento contém mais amostras que todos os outros, podemos dizer que seu erro quadrático médio é bastante relevante.

Para esse caso específico a alteração dos parâmetros destacados não resolveu nosso problema de modelagem, teremos então que para essa classe de ativo, fazer um tratamento diferente dos dados de entrada.

7.2.6 Conclusões sobre os ensaios

Dos ensaios realizados e apresentados resumidamente acima, podemos tirar algumas conclusões que nos guiarão até o fim do projeto. Uma delas é que algumas classes de ativos possuem comportamentos muito agressivos de variação de rentabilidade, o que nos força a buscar por um pré-processamento mais adequado do uma simples normalização. Porém para o caso de variáveis com comportamento mais suave como a cotação do Dólar, nosso pré processamento já é suficiente e a rede neural desenvolvida e apresentada no ensaio 2 já é suficiente.

Um outro fato importante é não se iludir com os resultados dos testes, uma vez que esse conjunto é muito restrito, é de maior interesse nos atentarmos ao erro nos conjuntos de validação e treinamento.

Cabe ainda destacar que nossa predição de concentração não necessita de uma altíssima precisão, apesar de sempre a desejarmos. Isto fica mais evidente se lembrarmos que o nosso cliente final tem o desejo de saber qual é a concentração por classe de ativo de um determinado gestor e com isso saber como ele está posicionado. Exemplificando: caso nosso cliente consulte o FEPREM e verifique que determinado fundo possui 40% de dólar no portfólio, com uma margem de erro de 5%, isso já é suficiente. Caso Nossa cliente achar que o dólar vai cair e que o fundo está mal posicionado, o dinheiro é retirado do fundo, não importa muito se a concentração é 35%, 40% ou 45%, o importa que a concentração de dólar é relevante dentro do fundo.

De uma forma geral obtivemos boas modelagens em 50% dos ensaios, alterando variáveis como função de transferência dos nós, passo de aprendizagem, quantidade de iterações. Para os casos que não obtivemos sucesso, em geral das

séries de dados mais voláteis, iremos agora nos aprofundar em pré-processamentos mais sofisticados nos dados de entrada.

7.3 Fundos reais

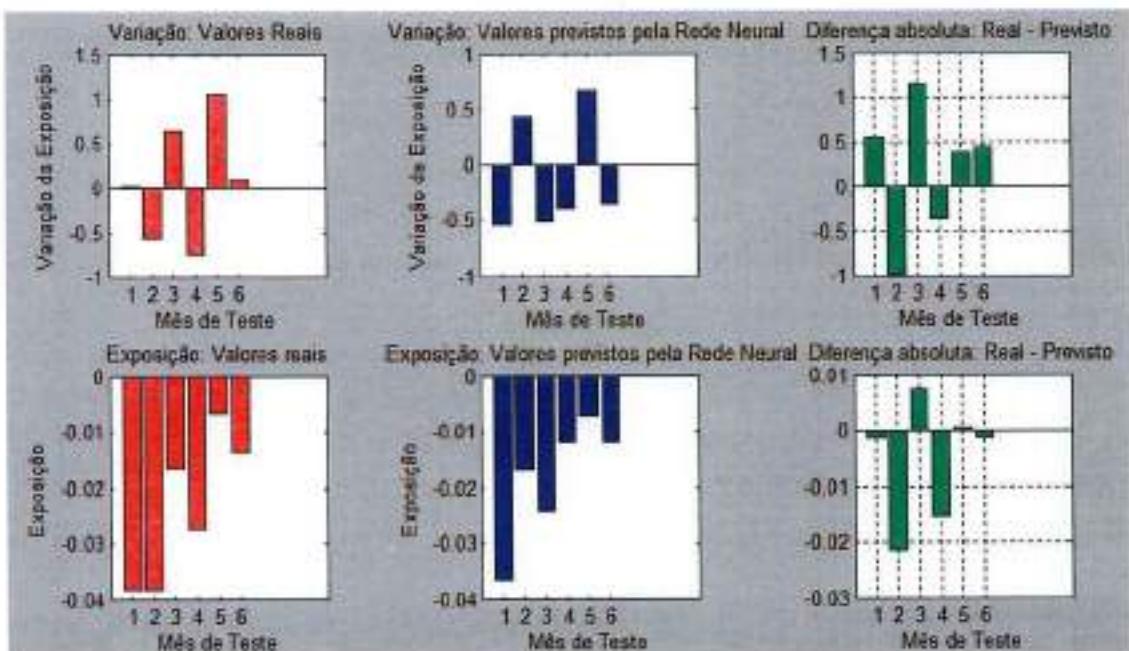
Evidenciaremos a seguir os cinco principais casos observados na modelagem das classes dos fundos.

7.3.1 Exposição variando num padrão complexo

Foi observado que nas vezes em que há uma variação com um padrão mais complexo na exposição da classe durante os seis meses de previsão, o modelo linear apresenta um erro muito maior que o modelo neural.

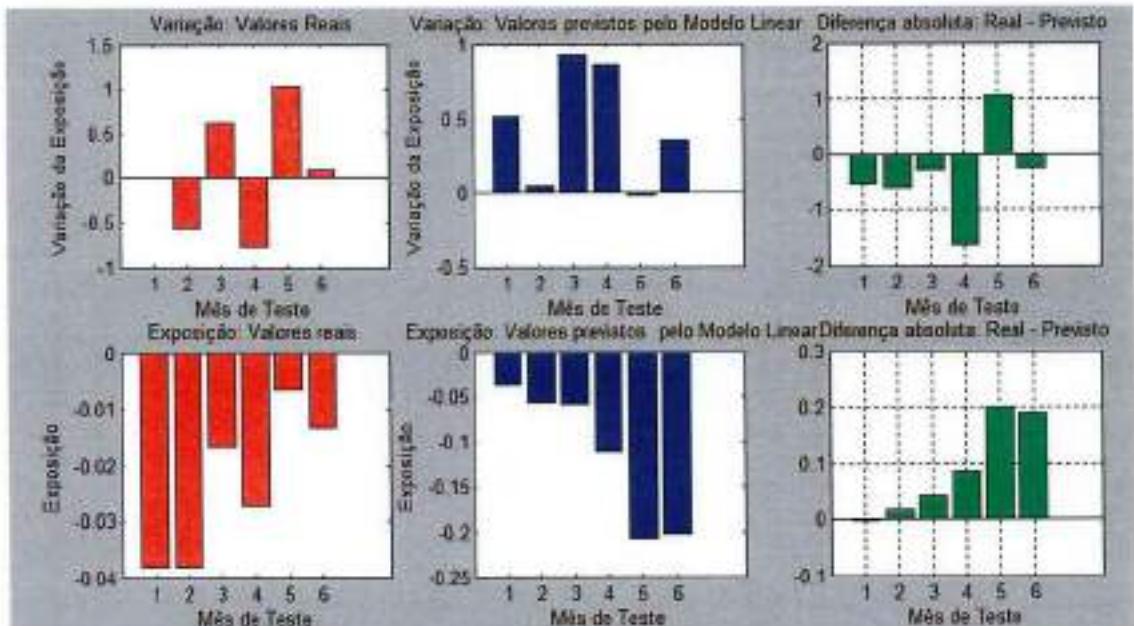
Este caso, que acontece em praticamente todos os fundos para algumas classes, é o que justifica o uso de redes neurais no produto. Podemos notar pela comparação abaixo que a variação é prevista de maneira muito satisfatória pela rede neural, ao passo que o modelo linear obtém um resultado insatisfatório. Nota-se também a alta não linearidade da saída, que no caso é a variação da exposição da classe.

Figura 40- Previsão pelo modelo neural do fundo NEO MULTI ESTRATÉGIA FIM para a classe Índice de Consumo



Fonte: Autor

Figura 41- Previsão pelo modelo linear do fundo NEO MULTI ESTRATÉGIA FIM para a classe Índice de Consumo



Fonte: Autor

Resultado dos testes:

Erro quadrático médio sobre o resultado final

Modelo neural: 0,000125

Modelo linear: 0,014209

Erro quadrático médio da rede neural

Modelo neural: 0,017572

Modelo linear: 0,026181

O erro quadrático médio sobre o resultado final obtido pelo modelo neural é duas ordens de grandeza menor do que o obtido pelo modelo linear. Tão importante quanto o erro baixo é, como pode ser observando pelos gráficos, a rede neural captar uma tendência de queda correta, ao passo que o modelo linear estima uma tendência de subida.

Este é, portanto, um exemplo no qual o resultado obtido pelo modelo auxilia plenamente o usuário na sua tomada de decisão em relação a sua exposição por classe de ativo. É também uma justificativa clara para o uso de modelagem não-linear no produto.

7.3.2 Exposição baixa e aproximadamente linear

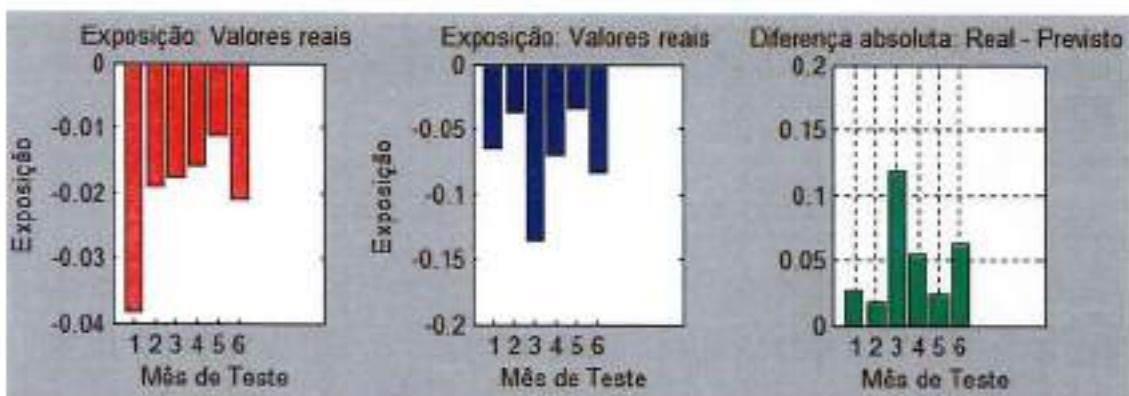
Nos casos em que a exposição possui um valor baixo (menor que 5%), geralmente observamos erros relativos maiores. Quando a variação da exposição foi mais próxima de um comportamento linear, o modelo linear obteve resultado semelhante ao neural.

Dado que o usuário do produto deseja apenas ter uma ideia de qual é a exposição daquele fundo para determinada classe, não sendo importante um alto grau de precisão, o erro relativo não é muito importante, e sim o erro absoluto.

Vejamos então um exemplo em que o comportamento do modelo linear possui um erro semelhante ao modelo não-linear. Como sabemos que a classe está

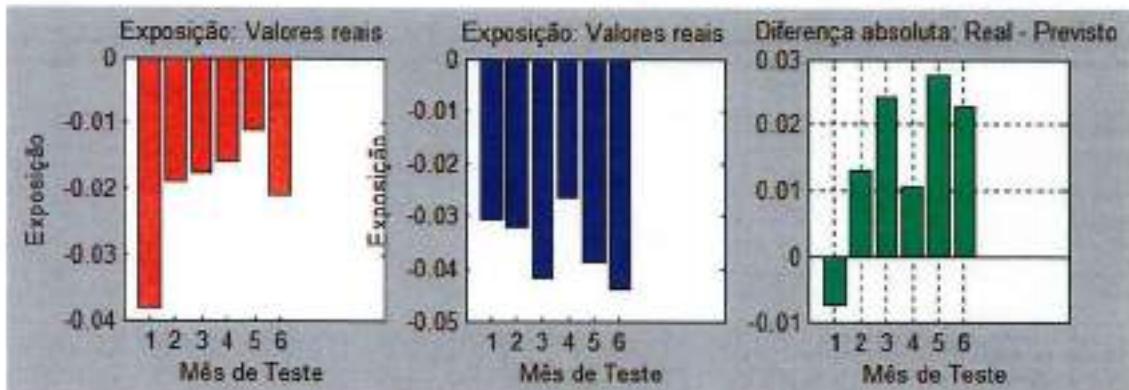
se comportando dessa forma, podemos optar por utilizar o modelo linear neste caso para o FEPREM.

Figura 42 - Previsão pelo modelo neural do fundo NEO MULTI ESTRATÉGIA FIM para a classe Dólar



Fonte: Autor

Figura 43 - Previsão pelo modelo linear do fundo NEO MULTI ESTRATÉGIA FIM para a classe Dólar



Fonte: Autor

Resultado dos testes:

Erro quadrático médio sobre o resultado final

Modelo neural: 0,000127

Modelo linear: 0,000216

Erro quadrático médio da rede neural

Modelo neural: 0,1031

Modelo linear: 0,0101

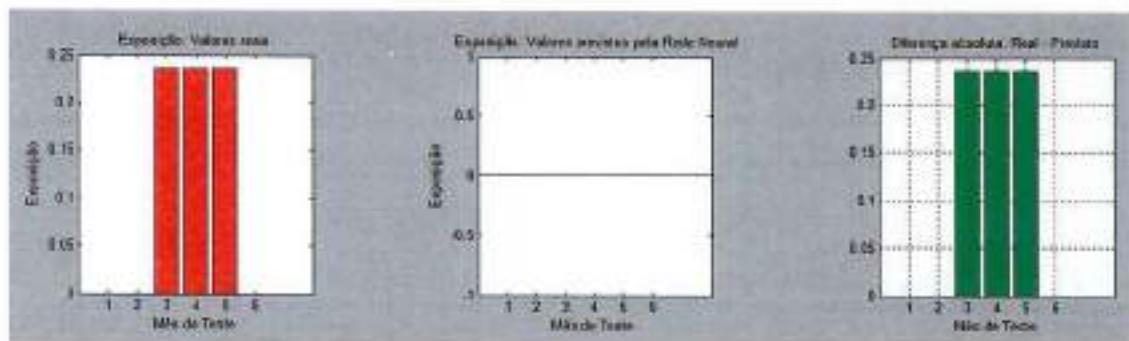
Podemos observar que os valores reais são pequenos em módulo, e que apesar dos erros relativos serem grandes, os erros absolutos são suficientemente pequenos. Também notamos que os desempenhos dos dois modelos são aproximadamente iguais, sendo portanto, um caso em que pode ser usado o modelo linear, por ser mais simples.

7.3.3 Variação imprevisível

De forma geral, os fundos de investimento tendem a manter um tipo de comportamento ao longo do tempo. Essa premissa é o que permite que façamos uma previsão sobre em que o fundo está investindo no momento. No entanto, dado que se trata de situações reais em que muitas variáveis estão envolvidas, é possível que um fundo mude subitamente de comportamento em relação à alguma classe de ativo.

Vejamos um exemplo onde o fundo possuía um valor de exposição em uma determinada classe igual a zero durante vários meses no conjunto de treinamento da rede (6 meses ou mais de atraso), mas houve uma mudança súbita para mais de 20% do patrimônio investido nesta classe.

Figura 44 - Previsão para o fundo NEO MULTI ESTRATEGIA FIM para a classe IMA-B5



Fonte: Autor

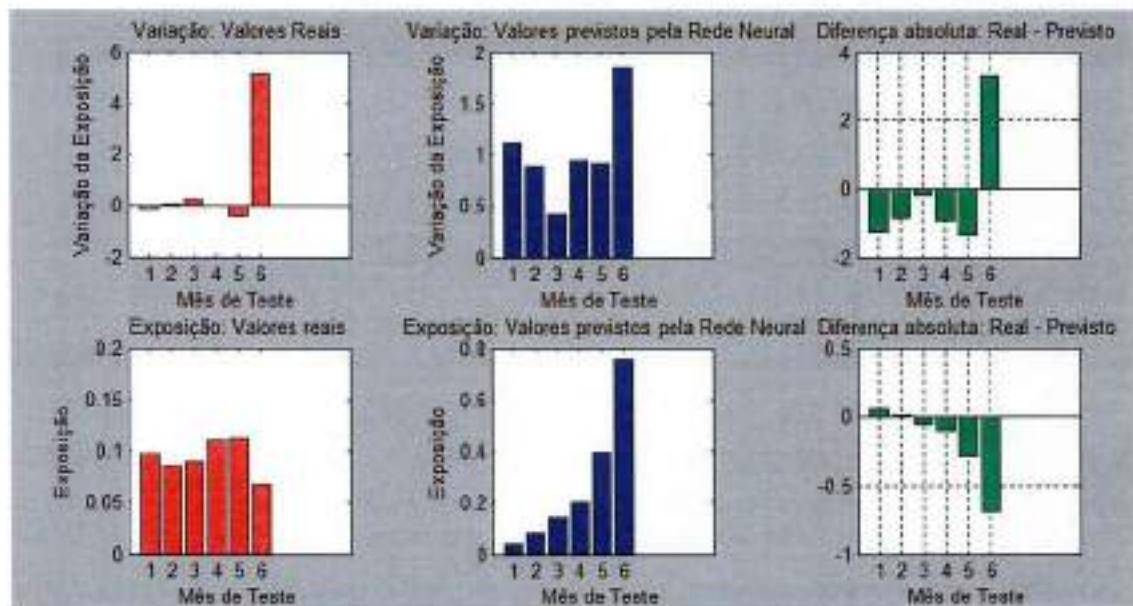
Em casos como este, tanto o modelo linear quanto o modelo neural são incapazes de realizar a previsão, pois não há evidências expostas nos dados de treinamento que permitem este tipo de previsão.

7.3.4 Modelo linear superior ao modelo neural

Em algumas raras vezes, pudemos observar que o modelo linear apresenta um erro menor que o erro do modelo neural. Isto acontece quando a magnitude da exposição é relativamente alta e se mantém aproximadamente constante.

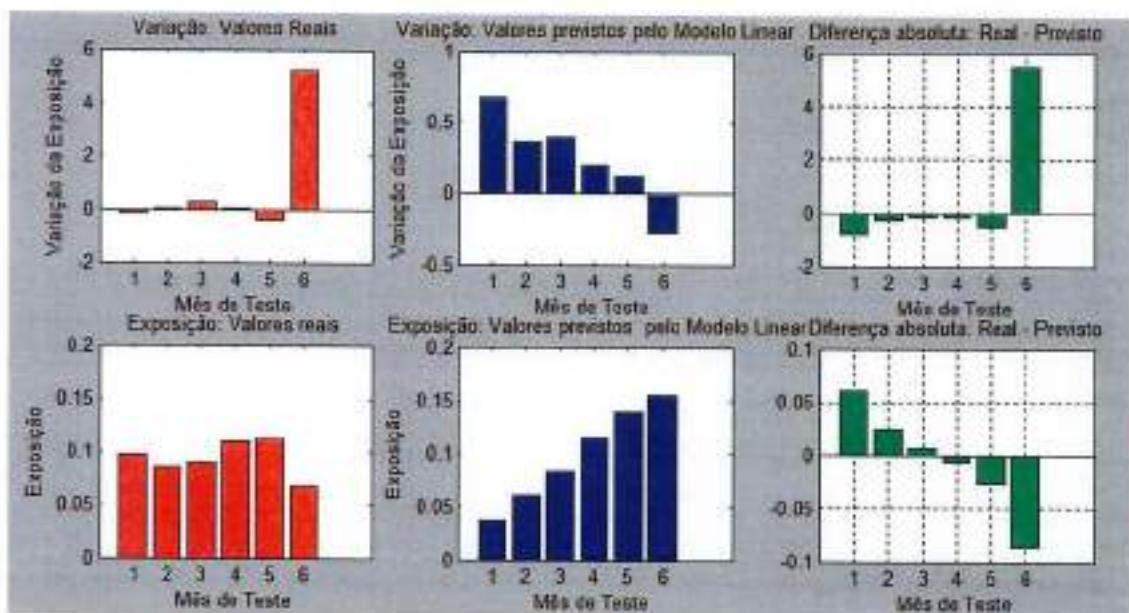
Nesse tipo de caso, observamos que o formato da curva nos dois modelos é parecida, mas o modelo linear conseguiu valores melhores de magnitude, como mostra o exemplo abaixo:

Figura 45- Previsão pelo modelo neural do fundo ITAU EQUITY HEDGE ADVANCED FIM para a classe Índice de Materiais Básicos



Fonte: Autor

Figura 46- Previsão pelo modelo linear do fundo ITAU EQUITY HEDGE ADVANCED FIM para a classe Índice de Materiais Básicos



Fonte: Autor

Resultado dos testes:

Erro quadrático médio sobre o resultado final

Modelo neural: 0,09465

Modelo linear: 0,00209

Erro quadrático médio da rede neural

Modelo neural: 0,04915

Modelo linear: 0,09519

Observamos que o modelo linear é a melhor alternativa para esta classe de ativo neste fundo, pois encontra valores de erro muito menores e o formato da curva mais preciso. É provável que exista parâmetros a serem usados para o modelo neural que melhorariam este resultado, e o tornaria próximo ou melhor que o linear, mas caso estes parâmetros não sejam encontrados para refinar o modelo, o modelo linear poderá ser utilizado.

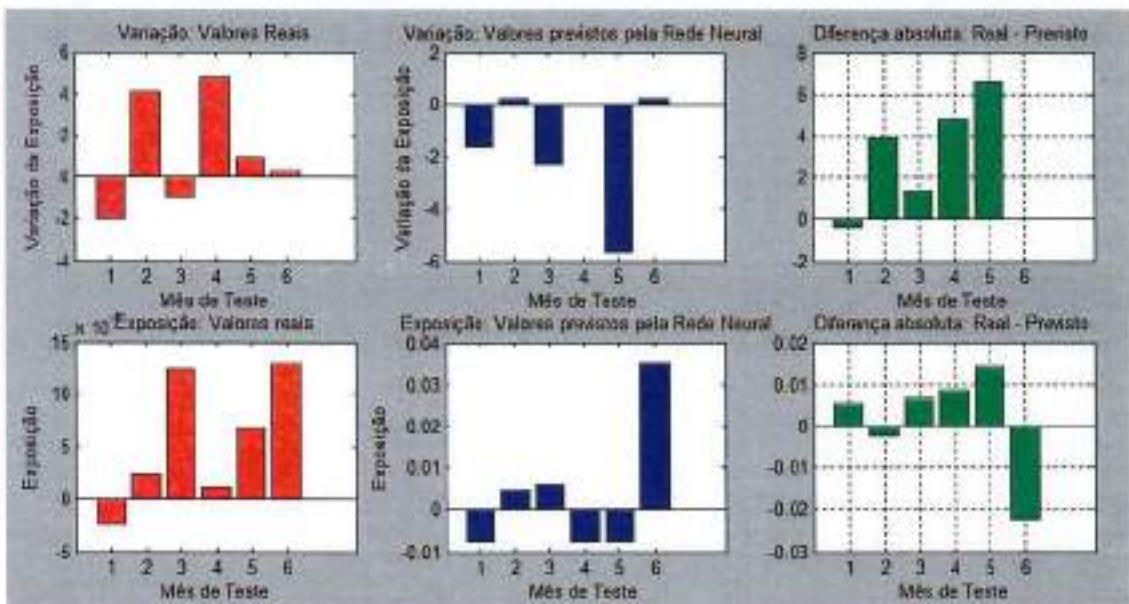
7.3.5 Fundos Long and Short

Fundos do tipo *Long and Short* possuem a característica de manter exposição baixa em todas as classes de ativos. Ao mesmo tempo em que fazem algum aumento de exposição em uma determinada classe, fazem uma operação de exposição negativa, de tal modo que a exposição resultante sempre é relativamente baixa. Este tipo de comportamento visa a diminuição de riscos (e consequentemente uma diminuição de retorno).

Não iremos nos aprofundar em que tipo de operações financeiras são feitas para obter exposição negativa, em que obtém-se lucro quando o valor do ativo cai, mas podemos ver um exemplo em que a exposição do fundo na classe em questão flutua constantemente perto do zero.

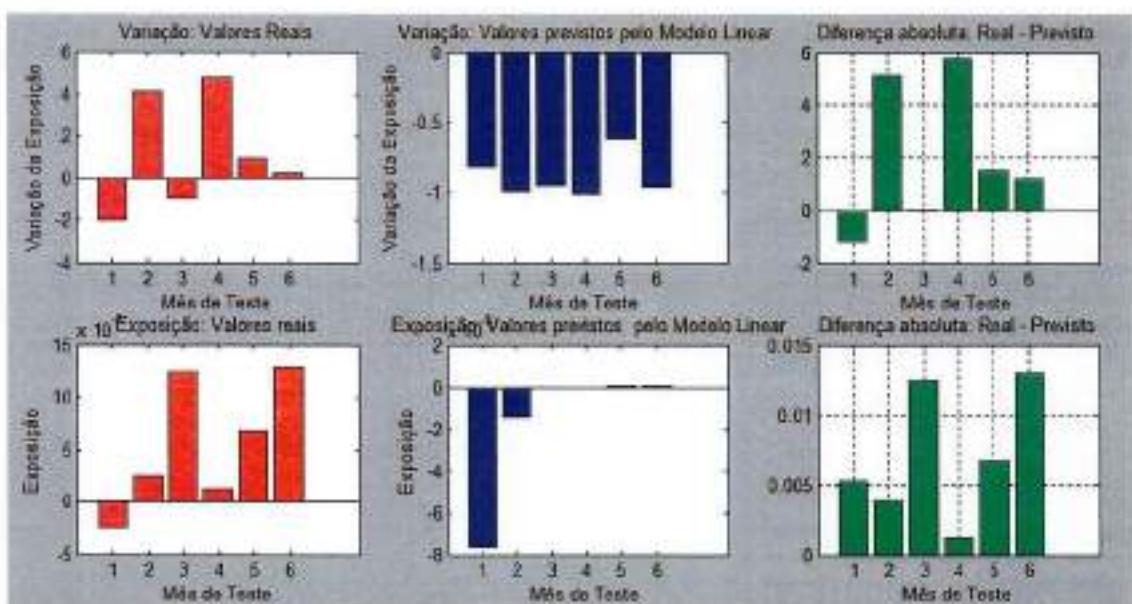
Nestes casos, assim como em todos com exposição baixa, apesar dos erros relativos serem baixos, os absolutos são pequenos, satisfazendo o objetivo de dar uma ideia da magnitude exposição total.

Figura 47 - Previsão pelo modelo linear do fundo BRZ LONG SHORT FIM para a classe Índice de Energia Elétrica



Fonte: Autor

Figura 48 - Previsão pelo modelo linear do fundo BRZ LONG SHORT FIM para a classe Índice de Energia Elétrica



Fonte: Autor

Resultado dos testes:

Erro quadrático médio sobre o resultado final

Modelo neural: 0,000141

Modelo linear: 0,000068

Erro quadrático médio da rede neural

Modelo neural: 0,1669

Modelo linear: 0,1302

Podemos observar que os erros quadráticos médios obtidos possuem a mesma ordem de grandeza nos dois modelos, com a diferença que o modelo neural consegue capturar melhor o formato da curva. Como os valores absolutos trabalhados são baixos, os dois modelos conseguem erros baixos, e como mencionado anteriormente, este sempre será o caso para fundos *Long and Short*, em que a política é sempre manter exposições baixas nas classes de ativo.

7.4 Análise quantitativa consolidada

Desenvolveremos a seguir uma análise de sete fundos multimercado para os quais realizamos o fluxo completo de projeto: Escolha de parâmetros, modelo da rede (entre os quatro apresentados anteriormente), exportação de dados e inclusão no site. No final desta seção será feita uma comparação entre os dois métodos adotados: Rede Neural e Modelamento Linear.

7.4.1 Rede neural

Primeiramente vamos apresentar os resultados obtidos com rede neural. Vemos na tabela abaixo uma compilação dos erros quadráticos médios gerados por fundo por classe.

Tabela 1 - Erros quadráticos médios dos fundo por classe - Redes Neurais

Classe	BNY MELLON ARX HEDGE FIM	SAFRA CURRENCY HEDGE 30 FIM	ITAU EQUITY HEDGE FIM	BTG PACTUAL HEDGE FIM	BRZ LONG SHORT FIM	HUMMER FIM	NEO MULTI ESTRATEGIA FIM	EQM da classe
IMA-B5	9,84E+00	-	2,37E-01	-	-	5,69E-02	2,79E-02	2,54E+00
IMA-B5+	1,84E+00	2,27E-06	2,45E+01	1,59E-06	4,12E-07	2,44E-01	1,09E-01	3,81E+00
IEE	1,52E-06	-	3,50E-07	1,68E-04	1,42E-04	1,78E-05	2,48E-05	5,90E-05
INDX	-	-	4,24E-06	-	1,45E-05	4,11E-07	5,04E-05	1,03E-05
ICON	-	-	1,83E-06	4,59E-03	1,87E-04	2,25E-04	1,25E-04	1,03E-03
IMOB	3,79E-06	-	6,88E-07	8,86E-05	2,38E-05	1,80E-05	1,09E-04	4,04E-05
IFNC	2,49E-06	-	1,37E-05	1,53E-03	1,11E-04	2,86E-05	4,16E-04	3,54E-04
IMAT	5,27E-06	-	9,60E-06	9,47E-02	5,88E-05	8,12E-05	4,36E-03	1,65E-02
UTIL	4,18E-06	-	3,32E-06	2,16E-03	3,61E-04	7,15E-05	1,28E-03	6,50E-04
Dólar	2,85E-03	1,90E-02	1,75E-05	-	-	2,58E-03	1,27E-04	4,92E-03
EQM do fundo	1,40E+00	9,52E-03	2,47E+00	1,47E-02	1,11E-04	3,04E-02	1,42E-02	

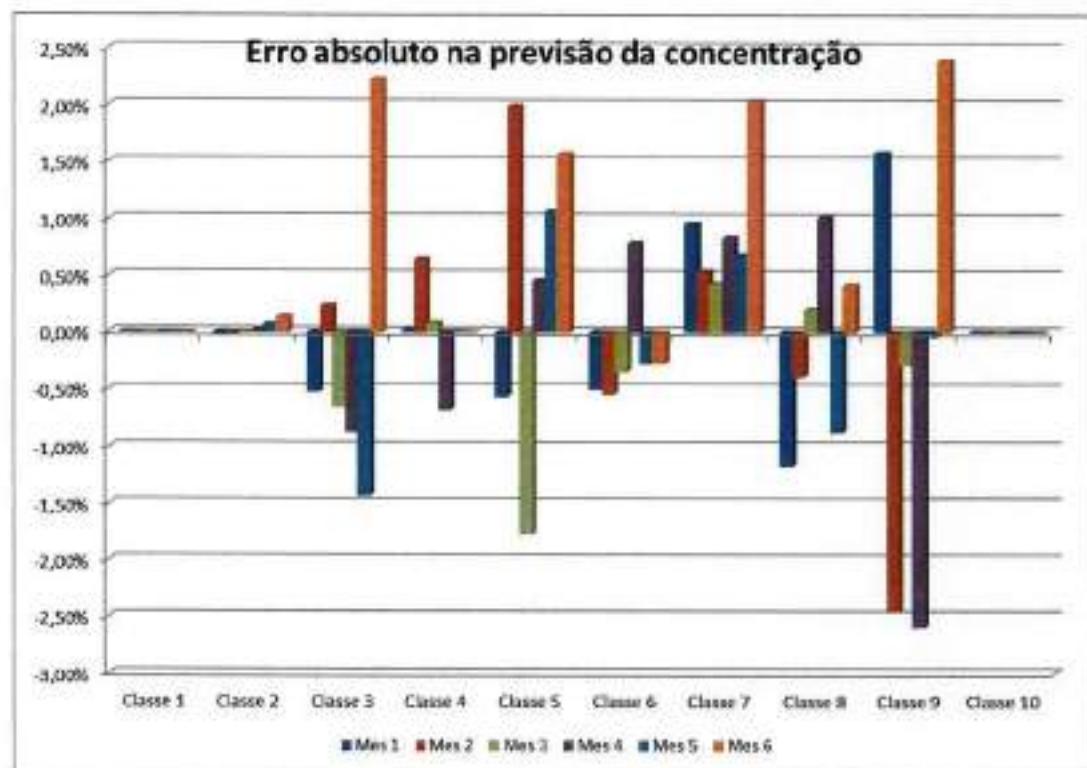
Fonte: Autor

Na ultima linha está a média dos erros quadráticos médios do fundo e na ultima coluna, a média dos erros quadráticos médios das classes. Podemos dessa forma verificar, de maneira geral, quais as classes e fundos melhores modelados. Os

erros apresentados acima são os do Sistema Final, ou seja, o das exposições por classe de ativo em si.

Como podemos observar, o fundo que conseguimos melhor modelar, foi o BRZ LONG AND SHORT FIM com média de erro da ordem de 10^{-4} , no gráfico abaixo que mostra a diferença entre valores calculados e reais (erro absoluto) verifica-se que apenas em poucos casos essa diferença foi superior a 2%. Concluímos que para esse fundo o modelamento por redes neurais é eficiente e satisfatório do ponto de vista do cliente. Já o fundo com pior desempenho de modelamento foi o BTG PACTUAL HEDGE FIM, que apresentou média de erros quadráticos médios das classes de 2,47.

Figura 49 - BRZ LOND AND SHORT - Erro final de previsão



Fonte: Autor

A média de erros quadráticos médios, foi a forma encontrada de representar a eficiência da modelagem em um único número. Ressaltamos que esse número não pode ser analisado de maneira isolada, é necessário verificar se em alguma de suas

classes obtivemos um erro altíssimo (situação imprevisível discutida na seção anterior). Isso de fato ocorre no caso do BTG PACTUAL HEDGE FIM, para classe IMAB-5+, que apresenta EQM de 24,5.

Quanto a classes de ativo, pode-se verificar que as classes IMAB-5 e IMAB-5+ são as que apresentam maior erro quadrático médio, da ordem de 1. Ressaltamos que a dificuldade de previsão dessas classes para os fundo BNY e BTG, contribui de forma decisiva no aumento dessa média. De modo geral todas as outras classes apresentaram baixo erro quadrático médio, de 55 classes com exposição diferente de 0, 11 apresentaram erro quadrático médio menor que 10^{-5} , ou seja 20%. Abaixo tabela que consolida essa informação.

Tabela 2- Percentual de fundos por ordem de grandeza de erro - Rede Neural

Erro Quadrático Médio	Classes que apresentam EQM igual ou inferior
1,0E-01	73%
1,0E-02	67%
1,0E-03	58%
1,0E-04	47%
1,0E-05	20%
1,0E-06	11%

Fonte: Autor

Mostraremos a seguir os resultados de erro obtidos pela rede neural em si, seja ela previsora de variação de exposição ou exposição em si. Estes números são de maior importância para confirmar a eficiência do modelamento, a análise foi feita com base nos dados de treinamento.

Tabela 3 - Erro quadrático médio das redes neurais.

Classe	BNY MELLON ARX HEDGE FIM	SAFRA CURRENCY HEDGE 30 FIM	BTG PACTUAL HEDGE FIM	ITAU EQUITY HEDGE ADVANCED FIM	BRZ LONG SHORT FIM	HUMMER FIM	NEO MULTI ESTRATEGIA FIM	EQM da classe
IMA-B5	4,11E-01	-	7,88E-02	-	2,85E-01	1,35E-01	2,22E-01	
IMA-B5+	1,29E-01	1,68E-01	1,32E-01	1,19E-01	3,54E-01	4,06E-01	1,54E-01	2,09E-01
IEE	4,18E-02	-	1,29E-01	4,84E-02	9,56E-02	2,97E-01	4,53E-02	1,09E-01
INDX	4,69E-02	-	9,49E-02	1,32E-01	3,07E-02	4,34E-01	1,97E-01	1,56E-01
ICON	-	-	4,45E-01	6,28E-02	2,03E-01	9,11E-02	1,87E-01	1,94E-01
IMOB	3,57E+00	-	1,13E-01	2,88E-02	2,16E-01	3,11E-01	7,83E-02	7,19E-01
IFNC	9,31E+00	-	1,32E-01	4,08E-02	1,84E-01	6,33E-02	6,59E-02	6,33E-01
IMAT	1,60E+00	-	1,02E-01	1,27E-01	2,60E-01	2,51E-01	1,76E-01	4,20E-01
UTIL	1,15E-01	-	1,52E-01	1,13E-01	1,04E-01	1,69E-01	9,91E-02	1,25E-01
Dolar	1,52E-01	7,58E-02	2,71E-01	2,83E-03	-	2,37E-01	1,33E-01	1,45E-01
EQM da fimada	1,04E+00	1,22E-01	1,69E-01	7,45E-02	1,61E-01	2,63E-01	1,25E-01	

Fonte: Autor

Diferentemente da analise anterior, os números acima evidenciam que o erro quadrático médio é praticamente constante por classe de ativo, contudo podemos fazer uma distinção quanto aos fundos melhores modelados: BNY MELLON ARX HEDGE FIM aparece como o com pior modelamento, dado a alta média dos erros quadráticos médios daquele fundo, e ITAU EQUITY HEDGE ADVANCED FIM aparece como o mais bem modelado. Para esses dois casos não houve um erro destoante, que deslocasse muito a média, diferentemente da analise feita para os erros do Sistema.

Assim como feito anteriormente, mostramos a seguir o percentual de redes por ordem de grandeza de erros.

Tabela 4 - Percentual de redes por ordem de grandeza de erro

Erro Quadrático Médio	Classes que apresentam EQM igual ou inferior
1,0E+01	83%
1,0E+00	78%
1,0E-01	24%
1,0E-02	2%
1,0E-03	0%

Fonte: Autor

Num primeiro momento, os erros acima podem altos, mas deve-se lembrar que o número apresentado ao cliente final possui um erro significativamente menor, pois estamos aqui prevendo o variação da exposição (na maioria dos casos), se a exposição é um número muito pequeno, uma variação mesmo com erro grande, gera erro absoluto pequeno, satisfazendo a necessidade do cliente. Além disso conforme dito anteriormente, fizemos também o modelamento linear e mostraremos a seguir os erros obtidos com a mesma metodologia utilizada nessa subseção, teremos então um parâmetro de comparação para os resultados da rede neural.

7.4.2 Modelagem linear

Vemos na tabela abaixo uma compilação dos erros quadráticos médios gerados por fundo por classe com a utilização do modelo linear.

Tabela 5 - Erros quadráticos médios dos fundo por classe - Modelamento Linear

Classe	BNY MELLON ARX HEDGE FIM	SAFRA CURRENC Y HEDGE 30 FIM	BTG PACTUAL HEDGE FIM	ITAU EQUITY HEDGE ADVANCED FIM	BRZ LONG SHORT FIM	HUMMER FIM	NEO MULTI ESTRATEGIA FIM	EQM da classe
IMA-B5	1,82E+01	-	1,53E+00	-	-	6,88E-02	2,79E-02	4,46E-00
IMA-B5+	6,81E+03	2,27E-06	3,86E+02	1,59E-08	2,07E-06	2,01E+00	1,43E+00	1,03E+03
IEE	1,79E-06	-	2,57E-07	1,46E-04	6,87E-05	1,75E-05	2,28E-05	4,29E-05
INDX	-	-	4,23E-03	-	1,45E-05	4,10E-07	8,07E-05	2,39E-05
ICON	-	-	1,83E-05	2,21E-02	1,03E-04	6,07E-05	1,42E-02	7,29E-03
IMOB	2,93E-06	-	6,87E-07	9,13E-05	2,05E-05	2,46E-05	9,47E-05	3,91E-05
IFNC	4,97E-04	-	1,38E-05	1,40E-03	1,32E-04	7,82E-05	1,35E-03	5,79E-04
IMAT	4,01E-05	-	9,60E-05	2,09E-03	1,12E-02	7,25E-05	4,03E-03	2,90E-03
UTIL	3,97E-06	-	3,32E-05	7,36E-01	2,10E-04	3,12E-05	1,32E-03	1,23E-01
Dólar	1,39E-03	3,38E-02	1,84E-05	-	-	2,15E-03	2,16E-04	7,51E-03
EQM do fundo	8,63E+02	1,89E-02	3,87E+01	1,09E-01	1,47E-03	2,04E-01	1,48E-01	

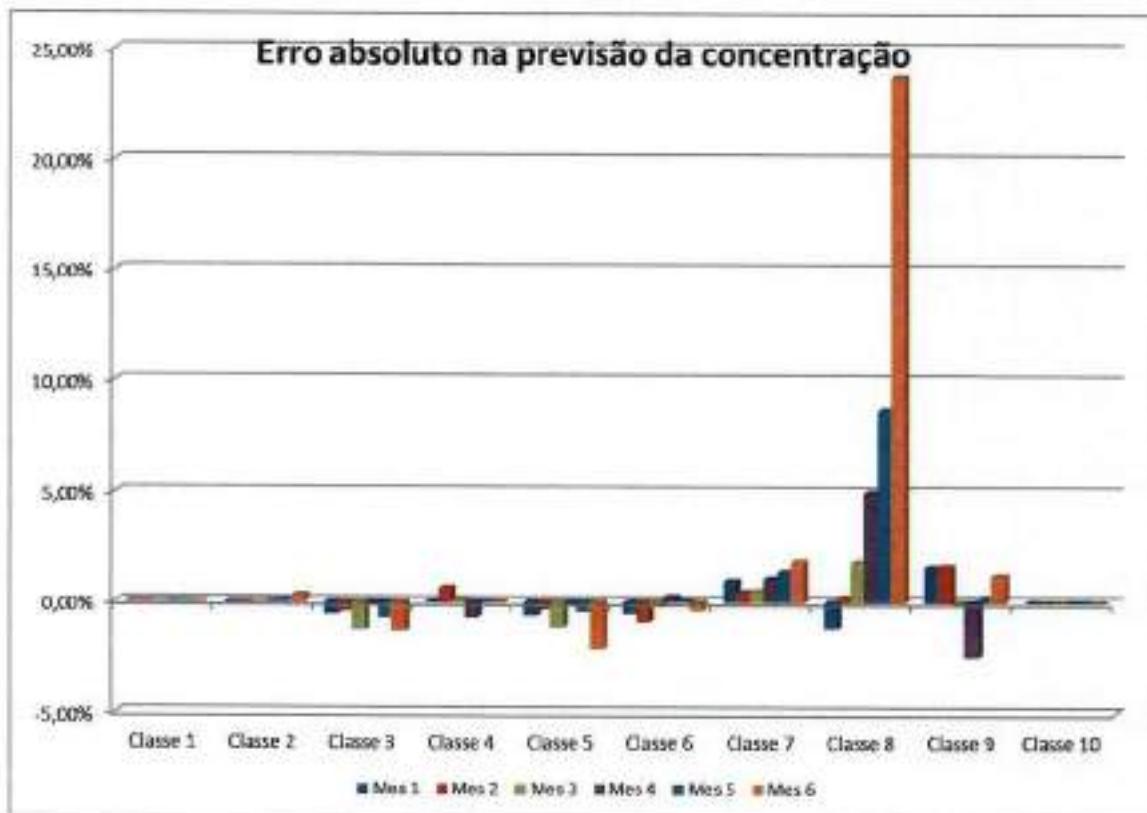
Fonte: Autor

Essa tabela tem a mesma estrutura da apresentada na subseção anterior. Destacamos aqui que diferentemente do caso em usamos rede neurais, o fundo com maior erro foi o BNY MELLON ARX HEDGE FIM. Já o com maior desempenho foi o BRZ LONG AND SHORT, assim como ocorreu com a modelagem por redes neurais.

Assim como no caso de modelamento por redes neurais, a previsão de exposição para classe IMAB-5+ (agora dentro do BNY) foi falha e fez com que a média dos erros quadráticos médios desse fundo fosse deslocada significativamente para cima.

Para efeito de comparação (subsecção posterior), segue abaixo o erro absoluto obtido para o fundo BRZ LONG AND SHORT.

Figura 50 - BRZ LOND AND SHORT - Erro final de previsão com modelo linear



Fonte: Autor

Conforme podemos observar há um ponto que apresentou erro significativamente maior que a média. Excluindo-se esse caso, observamos que em média o erro não supera os 5%.

Quanto ao erro do modelo linear por classe de ativo, segue na tabela abaixo um resumo por faixa de erro.

Tabela 6 - Percentual de fundos por ordem de grandeza de erro - Modelo Linear

Erro Quadrático Médio	Classes que apresentam EQM igual ou inferior
1,0E-01	71%
1,0E-02	64%
1,0E-03	56%
1,0E-04	47%
1,0E-05	20%
1,0E-06	9%

Fonte: Autor

Para concluir a apresentação de erros com método linear, mostramos a seguir os resultados de erro obtidos pelo modelamento em si, seja ele previsor de variação de exposição ou exposição em si.

Tabela 7 - Erro quadrático médio - modelamento linear

Classe	BNY MELLON ARX HEDGE FIM	SAFRA CURRENCY HEDGE 30 FIM	BTG PACTUAL HEDGE FIM	ITAU EQUITY HEDGE ADVANCED FIM	BRZ LONG SHORT FIM	HUMMER FIM	NEO MULTI ESTRATEGIA FIM	EQM da classe
IMA-B5	4,59E-01	-	1,21E-01	-	-	2,92E-01	6,00E-02	2,33E-01
IMA-B5+	1,87E-01	3,24E-01	2,08E-01	1,07E-01	3,36E-01	3,22E-01	1,64E-01	2,35E-01
IEE	3,57E-02	-	1,39E-01	4,86E-02	7,47E-02	3,34E-01	4,52E-02	1,13E-01
INDX	5,64E-02	-	6,49E-02	1,13E-01	8,18E-02	2,59E-01	7,01E-02	1,08E-01
ICON	-	-	2,04E-01	5,55E-02	1,21E-01	9,00E-02	1,26E-01	1,19E-01
IMOB	2,91E+00	-	1,11E-01	2,37E-02	1,61E-01	1,34E-01	3,29E-01	6,11E-01
IFNC	3,11E+00	-	2,54E-01	2,82E-02	1,20E-01	4,31E-02	6,55E-02	6,03E-01
IMAT	1,35E+00	-	9,31E-02	1,30E-01	1,53E-01	2,76E-01	1,25E-01	3,55E-01
UTIL	1,13E-01	-	1,46E-01	1,66E-01	1,24E-01	2,66E-01	1,15E-01	1,53E-01
Dólar	1,67E-01	1,02E-01	2,62E-01	5,15E-03	-	1,09E-01	1,28E-01	1,29E-01
EQM do fundo	9,33E-01	2,13E-01	1,00E-01	7,62E-02	1,45E-01	2,11E-01	1,28E-01	

Fonte: Autor

Como podemos observar, com o método linear o erro quadrático médio é praticamente constante por classe de ativo, contudo podemos fazer uma distinção quanto aos fundos melhores modelados: BNY MELLON ARX HEDGE FIM aparece como o com pior modelamento, dado a alta média dos erros quadráticos médios

daquele fundo, e ITAU EQUITY HEDGE ADVANCED FIM aparece como o mais bem modelado.

A seguir o percentual de redes por ordem de grandeza de erros.

Tabela 8 : Percentual de redes por ordem de grandeza de erro – Modelo Linear

Erro Quadrático Médio	Classes que apresentam EQM igual ou inferior
1,0E+01	83%
1,0E+00	78%
1,0E-01	22%
1,0E-02	2%
1,0E-03	0%
1,0E-06	0%

Fonte: Autor

7.4.3 Comparação entre os dois modelos

Em relação ao erro quadrático médios das modelagens em si, como pode observar pelas tabelas 8 e 4, em apenas 4% dos casos a rede neural obteve erro de uma ordem de grandeza menor, isso sugere que as diferenças da superioridade não são dessa ordem de magnitude. Com o objetivo de verificar o quão melhor é o modelamento por redes neurais, levantamos o número de casos em que o modelamento linear supera o neural. Os valores estão expressos na tabela a seguir:

Tabela 9 - Superioridade da modelagem linear

Nível de superioridade da rede linear	% de casos
20%	29%
30%	22%
40%	14%
50%	9%
60%	2%

Fonte: Autor

A tabela acima informa o percentual de casos em que a rede linear superou a neural acima de um certo patamar, conforme podemos observar, em apenas 14% dos casos a rede linear obteve erro quadrático médio 40% menor que o modelamento por redes neurais.

A título de ilustração, temos o caso de modelagem do fundo BRZ LONG AND SHORT, nas figuras 50 e 51 temos a previsão de exposição pelos métodos de Rede Neural e Modelagem Linear respectivamente, aqui fica claro a importância da rede neural, pode-se observar que além de gera um erro menor, a modelagem por rede neural não apresentou nenhum caso de erro significativo, diferentemente da modelagem por método linear, que apresenta erro de 25% na previsão de exposição na classe de ativo: Índice de Materiais no sexto mês de previsão.

8 INTERFACE COM O CLIENTE – SITE

Neste capítulo iremos mostrar as informações que serão apresentadas no site do nosso produto. No estado atual, todas as informações divulgadas no site são anteriores a 2013. Apenas para efeito de comparação, são mostradas também as exposições reais nos últimos seis meses do período analisado (julho a dezembro de 2012).

8.1 Tela de variação dos valores das cotas

A tela de variação dos valores das cotas, que pode ser visualizada na figura abaixo, contém as informações básicas de um determinado fundo, que é selecionado pelo cliente dentre os fundos disponíveis na plataforma. Explicando de maneira sucinta, esta tela fornece a evolução do valor da cota do fundo para um dado intervalo de tempo. O cliente poderá escolher entre os seguintes intervalos pré-estabelecidos: seis, doze e vinte e quatro meses. Assim que esta tela é exibida, o cliente vê, por padrão, a evolução da cota para doze meses.

Além disso ele terá a opção de acessar a tela de exposições do fundo escolhido, através de um link abaixo do gráfico das cotas.

Figura 51 - Tela de variação dos valores das cotas



Fonte: Autor

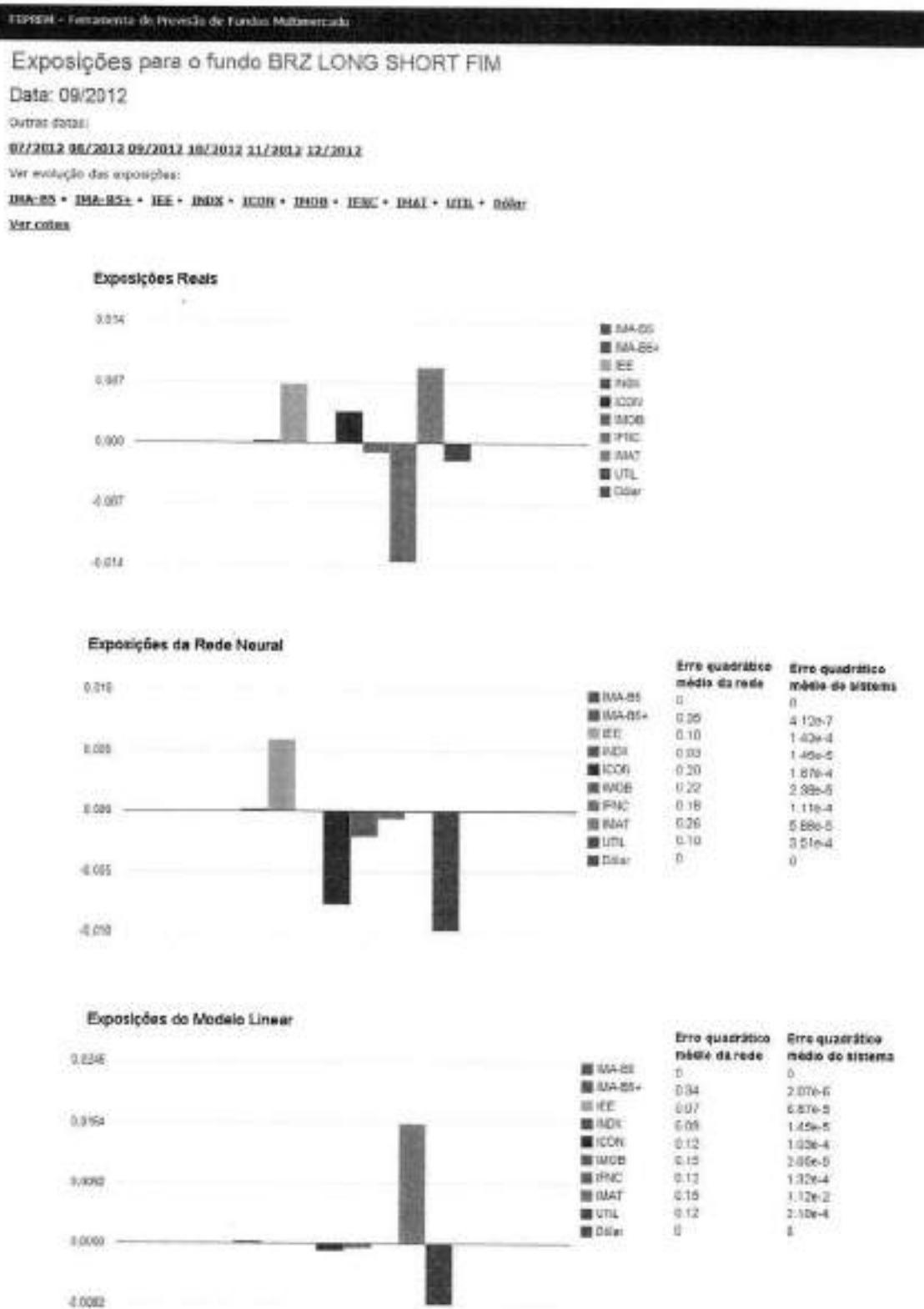
8.2 Tela de exposições

A tela de exposições, que está ilustrada na figura abaixo, mostra como foram feitas as alocações de um determinado fundo em relação a um conjunto de classes de ativos.

Existem três gráficos. O primeiro mostra a exposição real do fundo, como foi feita pelo próprio gestor. Esta informação está presente em nosso aplicativo somente em um período anterior a seis meses, dado que para datas mais recentes, essa informação não é pública. O segundo e terceiro gráficos mostram os resultados da rede neural e da modelagem linear respectivamente para os seis meses mais recentes.

O usuário pode mudar o mês para o qual as exposições são exibidas fazendo a seleção no menu superior. Além disso, pode acompanhar detalhes de cada exposição clicando no link correspondente. O menu em que constam os links para tais detalhes encontra-se logo abaixo do menu de datas.

Figura 52 - Tela de exposições de um fundo



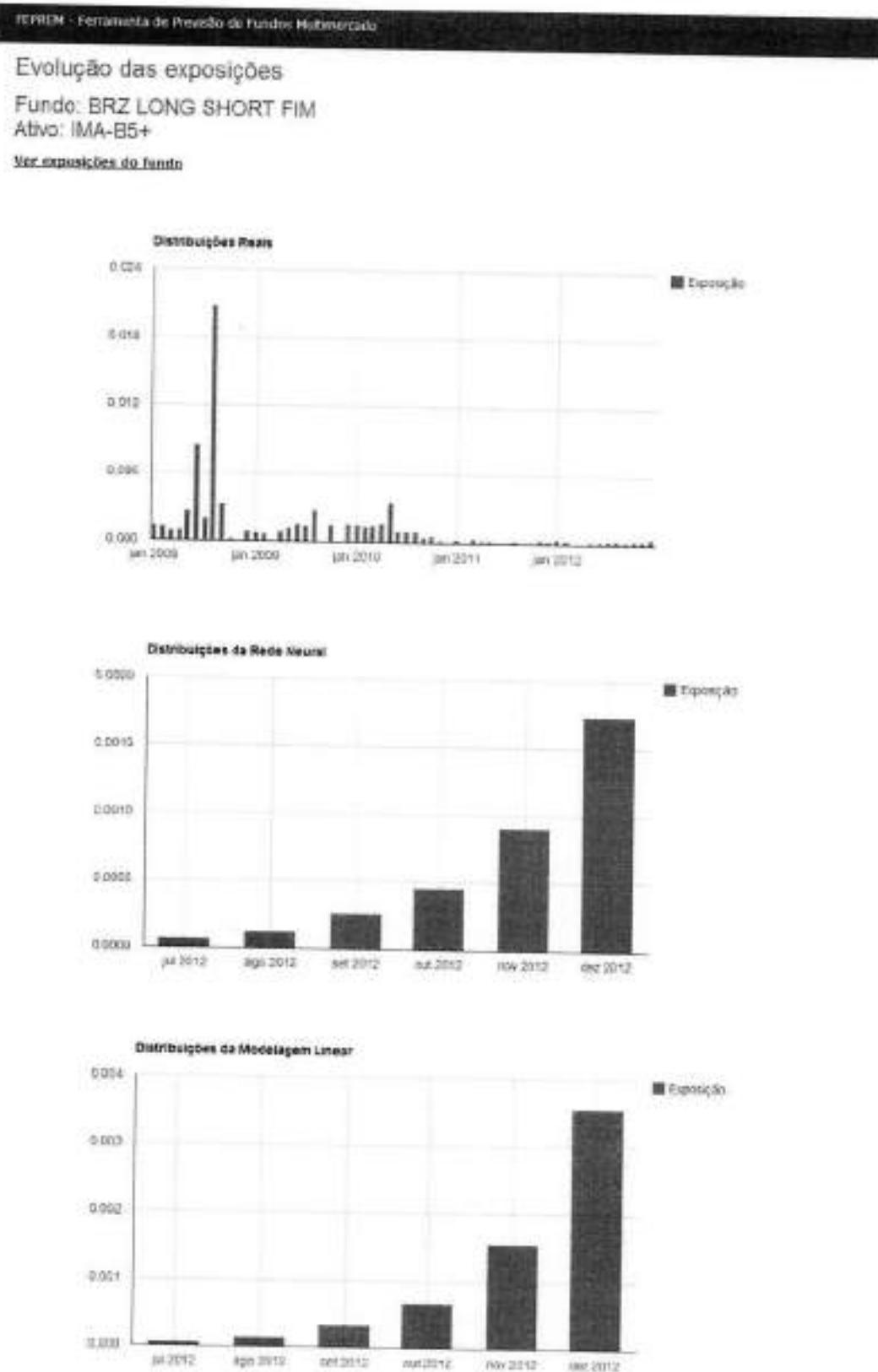
Fonte: Autor

8.3 Tela de evolução de exposição por classe de ativo

A tela de evolução das exposições por classe de ativo está detalhada na figura abaixo. Esta tela mostra como evoluiu a exposição de uma determinada classe de ativo para um determinado fundo. Mostra o período completo presente no banco de dados, ou seja, todos os dados disponíveis.

Assim como a tela de exposições, contém também três gráficos, um deles para os dados reais, outro para os resultados da rede neural e o último com resultados da modelagem linear.

Figura 53 - Tela de evolução das exposições



Fonte: Autor

8.4 Tela de cálculo de rentabilidade da carteira

Esta tela, ilustrada na figura abaixo, permite que o usuário faça uma simulação da rentabilidade de sua carteira. Os cálculos são feitos em tempo real e o valor da rentabilidade final é exibido logo abaixo.

O usuário deve preencher, para cada fundo, o tamanho de sua posição e também as datas de compra e venda. O sistema então acessa o banco de dados e retorna, para as datas selecionadas, as rentabilidades de cada fundo ponderadas pelo percentual preenchido pelo usuário. Em seguida soma estes resultados para obter a rentabilidade total da carteira.

Figura 54 - Cálculo de rentabilidade da carteira

FEPREM - Ferramenta de Previsão de Fundos Multimercado				
Fundo:	%	Data de Início:	Data Final:	Rentabilidade
BRZ LONG SHORT FIM	<input type="text"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	
HUMMER FIM	<input type="text"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	
ITAU EQUITY HEDGE AVD FIM	<input type="text"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	
BNY MELLON ARX HEDGE FIM	<input type="text"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	
SAFRA CURRENCY H 30 FIM	<input type="text"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	
BTG PACTUAL HEDGE FIM	<input type="text"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	1 <input type="button" value="▼"/> 2000 <input type="button" value="▼"/>	

Fonte: Autor

9 CONCLUSÃO

Concluímos que é válido e necessário o uso de redes neurais, pois como foi mostrado em secções anteriores, há casos de padrão complexo de variação de concentração, nos quais a rede linear não conseguiu prever o comportamento do gestor, e a rede linear previu com eficiência. Ainda é importante lembrar que para os números da tabela 9, estão os casos em que nenhum dos métodos foi capaz de prever a exposição, nesses casos a superioridade de uma rede em relação a outra possui pouco significado. Para os poucos casos em que o modelamento linear foi superior ao por redes neurais, e gerou bons resultados, o utilizaremos para prever os valores de exposição.

Ressaltamos também a importância da criação do fundo fictício para aprimoramento do modelo de redes neurais, pois permitiu que verificássemos quais variáveis são mais importantes no comportamento dos fundos. Pudemos notar que os fundos reais não-alavancados, realmente se comportam de forma semelhante ao fundo fictício.

Dada a grande quantidade de redes neurais que criamos, e consequentemente a grande quantidade de dados, foi fundamental a criação de um método automatizado de desenvolvimento de redes neurais, análise e tratamento estatístico de dados, permitindo um número muito maior de ensaios e a obtenção de modelos de maior qualidade.

REFERÊNCIAS

- ANDERSON, J.A. **An Introduction to Neural Networks**, Cambridge, MIT Press, 1995.
- BISHOP, C.M. **Neural networks for pattern recognition**, New York, Oxford University Press, 1995.
- Farhi, M. Derivativos financeiros: hedge, especulação e arbitragem. **Economia e Sociedade**, Campinas, v.13, p.93-114, dez. 1999.
- GATELY, E. **Neural networks for financial forecasting**, New York, John Wiley & Sons, 1995.
- McNELIS, P.D. **Neural networks in finance: gaining predictive edge in the market**, Burlington, Elsevier Academic Press, 2005.
- REFENES, A.P. **Neural networks in the capital market**, New York, John Wiley & Sons, 1995.
- TRYPHONAS M. **Modeling Hedge Fund Performance Using Neural Network Models**. Dissertation (Masters). University of Toronto. 2012.
- ANBIMA (SP). Disp em:<portal.anbima.com.br>. Acesso em: 21 nov. 2013.
- CakePHP cookbook v2.x Documentation** Disponível em:
<<http://book.cakephp.org/2.0/en/index.html>>. Acesso em: 5 mai. 2013.
- Fatores para a Escolha de uma Linguagem de Programação**. Disponível em: <<http://www.baguete.com.br/columnistas/columnas/51/paulo-krieser/07/05/2009/fatores-para-a-escolha-de-uma-linguagem-de-programacao>>. Acesso em: 6 mai. 2013.
- LINGUAGEM de Programação**. Disponível em:
<http://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o>. Acesso em: 5 mai. 2013.
- Qual Linguagem de Programação Escolher**. Disponível em:
<<http://www.devmedia.com.br/e-agora-qual-linguagem-de-programacao-escolher/15444>>. Acesso em: 6 mai. 2013.
- PHP and MySQL Web Development**. Disponível em: <<http://www.dba-24x7.com/articles/php-and-mysql-web-development-a-common-combination-with-rare-features.php>>. Acesso em: 7 mai. 2013.
- Processo de Desenvolvimento de Software**. Disponível em:
<http://pt.wikipedia.org/wiki/Processo_de_desenvolvimento_de_software>. Acesso em: 6 mai. 2013.