DIEGO GUERREIRO BERNARDES

**CARTEIRAS DE BLACK-LITTERMAN COM
ANÁLISES BASEADAS EM REDES NEURAI**

São Paulo
2019

DIEGO GUERREIRO BERNARDES

**CARTEIRAS DE BLACK-LITTERMAN COM
ANÁLISES BASEADAS EM REDES NEURAIS**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

São Paulo
2019

DIEGO GUERREIRO BERNARDES

**CARTEIRAS DE BLACK-LITTERMAN COM
ANÁLISES BASEADAS EM REDES NEURAIS**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Área de Concentração:
Engenharia de Sistemas

Orientador:
Oswaldo Luiz do Valle Costa

São Paulo
2019

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Bernardes, Diego Guerreiro

Carteiras de Black-Litterman com análises baseadas em Redes Neurais
/ D. G. Bernardes -- versão corr. -- São Paulo, 2019.

61 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1.Aprendizagem de Máquina 2.Redes Neurais 3.Otimização de Carteiras
4.Modelo de Black Litterman 5.Engenharia Financeira I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.

Às minhas mães e ao meu pai,
que me forneceram os blocos para
construir o meu caminho.

AGRADECIMENTOS

Professor Oswaldo, por ter apontado as direções e iluminado o caminho, sempre que os impasses surgiram.

Larissa, minha companheira, pela paciência e pela motivação.

Tio Haroldo, pelas discussões filosóficas e ajuda de contextualização deste trabalho na sociedade e no mundo.

Pedro Sugai, pela compreensão e pelo apoio.

Bruno Conti, pelas discussões que inicialmente motivaram este trabalho, e pelo auxílio nos temas relacionados à Redes Neurais.

À toda minha família, e especialmente à memória da minha tia Márcia, que foi a maior responsável por me propiciar a oportunidade de chegar onde cheguei.

E a todas as pessoas que participaram direta ou indiretamente deste processo.

“Você quer saber como ciborgues superinteligentes poderiam tratar humanos normais de carne e osso? É melhor começar a investigar como os humanos tratam seus primos animais menos inteligentes. A analogia não é perfeita, é claro, porém é o melhor arquétipo que podemos observar, e não só imaginar, no presente.”

-- Yuval Noah Harari

RESUMO

Neste trabalho é apresentado um sistema autônomo de gestão de carteiras que utiliza Redes Neurais Artificiais para monitoramento do mercado e o modelo de Black-Litterman para otimização da alocação de patrimônio. O sistema analisa as dez ações mais negociadas do índice Bovespa, com redes neurais dedicadas a cada ação, e prevê estimativas de variações de preços para um dia no futuro a partir de indicadores da análise técnica. As estimativas das redes são então inseridas em um otimizador de carteiras, que utiliza o modelo de Black-Litterman, para compor carteiras diárias que empregam a estratégia *Long and Short*. Os resultados obtidos são comparados a um segundo sistema de trading autônomo, sem o emprego da otimização de carteiras. Foram observados resultados com ótimo índice de Sharpe em comparação ao Benchmark. Buscou-se, assim, contribuir com evidências a favor da utilização de modelos de inferência bayesiana utilizados junto à técnicas quantitativas para a gestão de patrimônio.

Palavras-Chave – Engenharia Financeira, Redes Neurais, Otimização Não Linear, Inferência Bayesiana, Hedging (Finanças).

ABSTRACT

This work presents an autonomous portfolio management system which uses a Neural Network approach for monitoring the market and the Black-Litterman model for portfolio composition. The ten most traded assets from the Bovespa Index are analyzed, with dedicated neural networks, which suggests future return estimates using technical indicators as input. Those estimates are inserted in the Black-Litterman model which propose daily portfolio composition using long & short positions. The results are compared to a second autonomous trading system without the Black-Litterman approach. The results show great performance compared to the Benchmark, specially the risk and return relation, captured by the Sharpe Index. The work sought to bring positive evidences for the use of Bayesian Inference techniques in quantitative portfolio management.

Keywords – Financial Engineering, Neural Networks, Nonlinear Optimization, Bayesian Inference, Hedging (Finance).

SUMÁRIO

1	Introdução	10
2	Revisão Bibliográfica	12
3	Modelo de Black Litterman	15
3.1	Introdução: Os modelos de Black-Litterman	15
3.2	O Modelo de Referência	16
3.3	As variáveis Ω, τ e o modelo alternativo	18
4	Redes Neurais	20
4.1	Introdução: Um breve histórico da Inteligência Artificial	20
4.2	Neurônio Artificial	21
4.3	Conexionismo e Arquiteturas de Redes Neurais	23
4.3.1	Treinamento	24
4.3.2	A regra do Perceptron (Regra de Delta Simples)	26
4.4	Feed-Forward Multi-Layer Perceptron	26
4.4.1	O Algoritmo Back-Propagation	28
4.4.2	Funções de Ativação	29
5	Modelo Proposto e Metodologia	33
5.1	Primeiro Núcleo: Implantação das Redes Neurais	34
5.1.1	Indicadores da Análise Técnica	35
5.1.2	Conjuntos de Treinamento, Teste e Verificação	36
5.1.3	Seleção de Arquiteturas	37
5.1.4	Janelamento Móvel	39

5.1.4.1	Tamanho das Janelas	40
5.1.5	Treinamento das Redes Neurais e Saída do Primeiro Núcleo	42
5.2	Segundo Núcleo: Modelo de Black-Litterman	44
5.2.1	Matriz Ω e confiabilidade α	45
5.3	Sistema de Benchmark	47
5.4	Otimização de Carteiras	48
5.5	Súmula	48
6	Resultados	50
7	Conclusão	54
	Referências	56
	Anexo A – Demonstrações Matemáticas	59
A.1	Retorno esperado de Black-Litterman	59
A.2	Método de Theil	60

1 INTRODUÇÃO

O modelo de Black Litterman, estabelecido em Black e Litterman (1991, 1992) e busca reduzir a distância existente entre as aplicações práticas de otimização por média variância e sua proposta acadêmica, inicialmente feita por Markowitz (1952). He e Litterman (2002) e Michaud (1989) argumentam que não é incomum a otimização por média-variância sem restrições produzir carteiras impraticáveis e contra-intuitivas, com pouca diversidade, alta alavancagem e com alta sensibilidade aos erros nas estimativas de retorno esperado. A boa especificação de restrições, portanto, muitas vezes se mostra uma atividade que demanda bastante esforço na aplicação prática do modelo de Markowitz. A grande contribuição do modelo de Black-Litterman, entretanto, está na capacidade de combinar as interpretações de gestores de investimentos sobre o mercado com os retornos esperados do modelo de Markowitz, criando um mecanismo mais intuitivo para a construção de carteiras de ativos.

Esta abordagem é muito condizente com os objetivos buscados pelos sistemas de *soft-computing* aplicados à finança, como as técnicas de aprendizado de máquina aplicadas à predição de variações de preços de ativos financeiros, especialmente as redes neurais artificiais. Diversos trabalhos nas últimas década exploraram as potencialidades das redes neurais em sistemas preditivos no mercado financeiro. A proposta do sistema apresentado nesta dissertação é uma abordagem quantitativa do modelo de Black-Litterman utilizando *soft-computing*. Este sistema monitora o comportamento diário das dez ações de maior peso do índice bovespa (Ibovespa) e infere, a partir de um conjunto de redes neurais artificiais dedicadas, alimentadas por entradas baseadas em indicadores da análise técnica, previsões de retornos esperados. Estas previsões são inseridas em um núcleo construído a partir do modelo de Black Litterman, que sugere uma carteira casada de compra e venda (*Long & Short*). A performance do sistema é comparada com um segundo sistema de trading, também utilizando redes neurais construídas com a mesma metodologia, que negocia diretamente o índice Bovespa. Assim, busca-se contribuir na apresentação de evidências a favor do impacto positivo da utilização de inferência Bayesiana, como o Modelo de Black-Litterman, junto à técnicas quantitativas de gestão de patrimônio.

Esta monografia está organizada da seguinte maneira: No capítulo 2 é apresentada uma breve revisão bibliográfica de trabalhos relacionados ao tema. O capítulo 3 apresenta o modelo de Black-Litterman, as variações mais utilizadas na literatura e a base teórica utilizada. Em seguida, no capítulo 4, apresentam-se um breve histórico e alguns conceitos fundamentais de inteligência artificial, machine learning e redes neurais artificiais, com especial atenção ao Perceptron Multicamadas com alguns detalhes do seu aspecto construtivo, bem como as técnicas de treinamento utilizadas neste trabalho. No capítulo 5 é descrito de maneira detalhada o funcionamento do sistema proposto, alguns resultados parciais e como foi feita a junção dos dois núcleos do sistema. Finalmente, no capítulo 6, são apresentadas as métricas de desempenho do sistema e os resultados comparativos com o *Benchmark* seguida de uma breve discussão sobre os resultados obtidos e sugestões de aprimoramento do sistema no capítulo 7.

2 REVISÃO BIBLIOGRÁFICA

Black e Litterman (1991) e Black e Litterman (1992) estabelecem as bases conceituais do que posteriormente viria a ser conhecido como o modelo de Black-Litterman. O modelo é uma proposta complementar às técnicas de otimização por média-variância que, segundo os autores, apresentam dificuldades de implementação que as distanciam da realidade do mercado.

A proposta dos autores utiliza o conceito de “visões do investidor” que, somadas aos retornos esperados de equilíbrio de uma carteira de referência, produz uma nova carteira otimizada a partir dos conceitos de covariância da técnica de Markowitz (1952). As visões do investidor compõem um mecanismo por onde o administrador de um portfolio pode manifestar sua interpretação sobre as tendências do mercado, ou combinar as interpretações de analistas, utilizando um conjunto de matrizes. Ao se aplicar a otimização reversa, técnica derivada a partir da otimização de carteiras de Sharpe (1964), obtém-se uma aproximação dos retornos utilizados naquela otimização, que podem ser considerados retornos de equilíbrio.

O mecanismo de combinação entre a carteira de equilíbrio e a carteira de visões do investidor utiliza um parâmetro de confiança para cada expressão de visão. Ou seja, na ausência de visões, ou caso o investidor não possua uma opinião formada sobre um ou mais ativos, é utilizado o retorno de equilíbrio para composição da carteira, que se torna a própria carteira de equilíbrio. O desvio entre o retorno de equilíbrio e o retorno *a posteriori*, chamado de retorno de Black-Litterman, relaciona-se com o nível de confiança, ou erro, que o investidor apresenta para sua estimativa. Entretanto, as publicações originais do modelo não apresentam todo o desenvolvimento matemático para sua implantação, apresentados em trabalhos posteriores, fato este que poderia explicar o baixo número de trabalhos relacionados ao modelo ao longo da década de 1990.

A influência do modelo de Black-Litterman aumentou nas décadas seguintes, em grande parte devido a outros trabalhos ao final da década de 1990 e começo da década de 2000, como He e Litterman (2002), Satchell e Scowcroft (2000), Lee (2000) e Idzorek

(2007) que apresentam revisões e implementações do modelo original, bem como desenvolvimentos complementares, propondo alternativas de abordagem no cálculo de alguns parâmetros, e mostrando a intuição por trás dos conceitos utilizados na construção do modelo.

Idzorek (2007), além de apresentar uma implementação passo a passo do modelo, propõe uma abordagem mais intuitiva para o cálculo da matriz de variância Ω , utilizando um índice de confiança entre 0% e 100% para cada visão do investidor.

Fabozzi, Focardi e Kolm (2006) implementa uma estratégia de *trading* utilizando o modelo de Black-Litterman. É reforçado que muitas das estratégias de *trading* utilizadas hoje não podem ser facilmente traduzidas em previsões de retornos esperados como requer a otimização por média-variância clássica, ou mesmo o modelo de Black-Litterman. O maior desafio em se utilizar o modelo de Black-Litterman com estratégias de *trading* está em traduzir as interpretações subjetivas de performance futura dos ativos para uma visão quantitativa necessária ao modelo. O sistema proposto por Fabozzi utiliza-se da estratégia de momentum, proposta por Jegadeesh e Titman (1993), em conjunto com a análise transversal, e classifica em um ranking os ativos que compõem o índice global MSCI de países desenvolvidos. A estratégia aloca pesos positivos (compra) na matriz P para os ativos que pertencem à metade superior do ranking, composta por ativos com melhor retorno nos últimos nove meses, e pesos negativos (venda) aos ativos da metade inferior, com pior retorno nos últimos nove meses, ponderados pela volatilidade de cada ativo para evitar uma grande alocação em ativos de risco. A esta carteira aplica-se a técnica de otimização reversa, e seus retornos são então combinados aos retornos esperados de equilíbrio utilizando-se o modelo de Black-Litterman. Ao final de cada mês é feito um rebalanceamento da carteira final e calculados os indicadores de desempenho daquele mês.

A utilização de técnicas de aprendizagem de máquina na área financeira, com o intuito de facilitar, aprimorar ou substituir a tomada de decisões dos investidores acompanhou a revolução da computação ao longo das últimas décadas. É notável o impacto de trabalhos da década de 1980 e início da década de 1990 como Kimoto et al. (1990), White (1988) e Kaastra e Boyd (1996), estabelecendo alicerces importantes para trabalhos posteriores expandido as aplicações de técnicas com redes neurais.

Klassen (2007) investiga o uso de indicadores técnicos em um sistema de predição do preço de fechamento do índice *NASDAQ*, com destaques no uso de médias móveis exponenciais e do índice MACD (*Moving Average Convergence Divergence*). Martinez et al. (2009) usa um sistema de trading com redes neurais, utilizando médias móveis

exponenciais e Bandas de Bollinger, para prever pontos de entrada e saída intradiárias (*day-trade*) em ações da Bovespa. Maciel e Ballini (2010) utiliza valores históricos de preços de fechamento de índices de mercados ao redor do mundo e compara a performance de previsão de preço de fechamento entre redes neurais artificiais e modelos GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*). Gambogi e Costa (2013) constroem um sistema de trading seguidor de tendência, incorporando conceitos do efeito momento e a técnica de janelamento móvel, utilizando os indicadores técnicos MACD e ROC (*Rate Of Change*), juntamente com dados históricos de fechamento.

Embora muitos trabalhos utilizando o modelo de Black-Litterman em gestão ativa de carteiras foram encontrados, poucos destes trabalhos utilizam sistemas de redes neurais artificiais em conjunto com esta abordagem, proposta aqui apresentada. De maneira bastante inovadora, o trabalho de Xing, Cambria e Welsch (2018) apresenta um sistema composto de redes neurais para monitoramento de opinião pública através de redes sociais na composição de visões de mercado utilizadas no modelo de Black-Litterman. Em Wutsqa e Kusumawati (2016) e Zimmermann, Neuneier e Grothmann (2001) são apresentados modelos de previsão de preços para composição das visões de Black-Litterman, assim como neste trabalho, porém sem a utilização de indicadores de análise técnica.

3 MODELO DE BLACK LITTERMAN

3.1 Introdução: Os modelos de Black-Litterman

O modelo de Black-Litterman estima a distribuição dos retornos de uma carteira utilizando inferência Bayesiana com informações parciais. Para possibilitar a construção do modelo, assume-se que os retornos dos ativos comportam-se como variáveis aleatórias com distribuições normais de média desconhecida e variância conhecida. Algumas liberdades de implementação do modelo podem ser encontradas na literatura, sendo duas delas mais recorrentes. A implementação original, de Black e Litterman (1992) e He e Litterman (2002) por vezes chamada de modelo de referência, e uma implementação utilizando abordagem de encolhimento (*Shrinkage*) observada nos modelos de Satchell e Scowcroft (2000) e Meucci (2006), aqui chamada de modelo alternativo.

A teoria moderna de carteiras, consagrada no trabalho de Markowitz (1952), considera o retorno (r) de um ativo como uma variável aleatória com distribuição de probabilidade gaussiana com valor esperado (μ) e variância (Σ) — $r \sim N(\mu, \Sigma)$. O modelo de referência considera o valor esperado (μ) de r também uma variável aleatória, de distribuição gaussiana desconhecida passível de inferência. Na inferência bayesiana parte-se de uma premissa inicial, conhecida *a priori*, e infere-se um novo valor *a posteriori* a partir de informações adquiridas ou evidências encontradas. Para a estimação de μ parte-se do valor de equilíbrio calculado pelo método CAPM (*Capital Asset Pricing Model*), definido em Sharpe (1964), e agrega-se as informações das crenças — subjetivas ou quantitativas — dos agentes do mercado, chamadas de visões dos investidores no trabalho original, como novas informações. O modelo de Black-Litterman assume, portanto, que o preço de um ativo é definido por duas fontes de informações — o prêmio de risco no equilíbrio de mercado, e as visões dos investidores.

No equilíbrio de mercado, uma abstração matemática, todas as demandas e ofertas por um ativo são igualadas e os investidores que mantêm um ativo esperam um retorno definido pelo prêmio de risco de equilíbrio (Π). Uma maneira de estimar o prêmio de risco

é assumir que todos os agentes de mercado utilizam a metodologia do CAPM. Assim, pode-se calcular o prêmio de risco através de uma abordagem inversa à do CAPM.

O cálculo do valor de μ parte deste conceito. O valor *a priori* de μ – uma variável aleatória não observável – seria, portanto, o valor do prêmio de risco de equilíbrio mais um valor de erro de estimativa oriundo da incerteza sobre o CAPM e o equilíbrio. Ou seja:

$$\mu = \Pi + \epsilon^e$$

Com ϵ^e sendo uma variável aleatória de média igual a zero e variância igual a $\tau\Sigma$. Pode-se notar que τ , neste caso, é um escalar que denota a incerteza em cima da estimativa *a priori*, ou a confiança no CAPM – incerteza esta de alguma maneira proporcional à sua variância Σ . É justamente esta variável τ que protagoniza a principal diferença entre os modelos de referência e alternativo. No modelo de referência o valor esperado do retorno de um ativo segue uma distribuição gaussiana dada por $\mu \sim N(\Pi, \tau\Sigma)$.

O modelo alternativo considera o valor esperado μ não como uma variável aleatória com distribuição gaussiana, mas como uma estimativa pontual. Implica-se, então, que a informação *a priori* da inferência Bayesiana iguala-se ao prêmio de risco de equilíbrio, ϵ^e iguala-se a zero e a variável τ , ainda presente, adquire um significado de variável de controle no modelo. Neste novo contexto a variável τ ajusta quanto as visões do investidor devem influenciar no valor *a posteriori* do retorno do ativo, ou seja, ela pode ser considerada como uma variável de confiabilidade nas visões dos investidores. Como observado por Walters (2014), existem algumas implicações nesta interpretação de τ , discutidas na seção 3.3.

3.2 O Modelo de Referência

Na equação (3.1), equação de referência do modelo de Black-Litterman, o retorno *a posteriori* estimado possui uma distribuição normal de média $\hat{\Pi}$, com variância M obtida pela equação (3.2). Tanto pelo método de estimação mista de Theil, Theil (1971), quanto pelo teorema de Bayes, pode-se chegar à estas equações. A dedução pelo método de Theil pode ser encontrada no apêndice A.2.

$$\hat{\Pi} = [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P'\Omega^{-1}Q] \quad (3.1)$$

$$M = \Sigma + [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1} \quad (3.2)$$

Nestas equações as informações *a priori* são expressas na forma da matriz de covariância $\tau\Sigma$ e no prêmio de risco de equilíbrio Π .

O prêmio de risco Π pode ser obtido pela otimização reversa da função utilidade quadrática do investidor – que resulta em $\Pi = \delta\Sigma w$, com w sendo o conjunto de pesos de uma carteira e δ o coeficiente de aversão ao risco desta carteira.

As informações parciais sobre o mercado são representadas pelas matrizes P , Q e Ω , que compõem as visões dos investidores. Estas visões são impressões, crenças ou constatações quantitativas do investidor sobre o mercado e são expressas como retornos esperados. Estes retornos podem ser na forma absoluta, onde o investidor prevê a oscilação de um ativo, na forma relativa, onde o investidor prevê a oscilação de um ativo em relação a outro(s), ou uma combinação destas situações.

Seja um mercado de n ativos, onde expressam-se k visões, as matrizes das visões do investidor podem ser entendidas na forma de uma carteira, cujo retorno pode ser descrito pela equação (3.3).

$$P\mu = Q + \epsilon^\nu \quad (3.3)$$

Sendo μ , um vetor com dimensões $n \times 1$, o retorno esperado da carteira que as visões exprimem – outra interpretação de μ é a informação condicional parcial da inferência Bayesiana, em forma de retorno esperado –, P a matriz de pesos, dimensões $k \times n$, que compõe a carteira de visões, Q a matriz de estimativas de retornos do investidor, de dimensões $k \times 1$, e ϵ^ν indica a incerteza nas estimativas de Q . Caso o investidor expresse uma visão na forma absoluta, a soma dos pesos da linha da matriz P para esta visão deve ser igual à 1, e para a forma relativa a soma dos pesos deve ser igual à 0.

O componente ϵ^ν , neste caso, pode ser interpretado como a incerteza na especificação da visão. Este pode representado por um vetor de ruído branco de média zero e variância finita e conhecida, igual à Ω , de dimensões $k \times k$. Embora não seja uma exigência matemática, costuma-se expressar os componentes da matriz Ω (ω_i) como independentes e não correlacionadas, tornando-a uma matriz diagonal. Esta simplificação ajuda na especificação de Ω pois, caso contrário, o investidor teria que estimar não só a incerteza na sua expectativa como também a incerteza na correlação entre as suas expectativas, incorrendo em uma atividade bastante abstrata.

Por exemplo:

Considere um mercado de quatro ativos onde o investidor deseja manifestar duas opiniões. A primeira opinião, manifestada de forma relativa, diz que o ativo 1 terá uma performance 2% melhor que o ativo 3, com uma confiabilidade ω_1 . A segunda opinião, expressa em forma absoluta, diz que o ativo 2 irá desvalorizar 6%, com uma confiabilidade ω_2 . Desta forma, temos as três matrizes a seguir:

$$P = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 & 0 \end{bmatrix}; Q = \begin{bmatrix} 3 & -6 \end{bmatrix}; \Omega = \begin{bmatrix} w_1 & 0 & w_2 \end{bmatrix}$$

3.3 As variáveis Ω , τ e o modelo alternativo

A equação (3.1) do retorno esperado *a posteriori* pode ser reescrita na equação (3.4), onde é possível observar como as visões dos investidores formam um segundo componente que é adicionado ao retorno esperado *a priori*. A demonstração desta transformação pode ser acompanhada no apêndice A.1

$$\hat{\Pi} = \Pi + \tau \Sigma P' [(P \tau \Sigma P') + \Omega]^{-1} [Q - P \Pi] \quad (3.4)$$

Analisando a equação (3.4) nota-se que para $|\Omega| \rightarrow 0$, o retorno *a posteriori* se iguala aos da carteira de visão do investidor – equação (3.3) – e, para $|\Omega| \rightarrow \infty$, estes retornos se igualam ao prêmio de equilíbrio Π . Ou seja, Ω é uma grandeza inversamente proporcional à confiança do investidor em suas estimativas, e pode ser interpretado como a covariância das visões do investidor. A abordagem para o cálculo de Ω é aberta à discussão. Em He e Litterman (2002) considera-se que esta covariância é proporcional à covariância *a priori*, como segue:

$$\begin{aligned} \omega_{ij} &= p_i(\tau \Sigma) p_j^T, \quad \forall i = j \\ \omega_{ij} &= 0, \quad \forall i \neq j \\ \text{ou} \\ \Omega &= \text{diag}(P(\tau \Sigma) P') \end{aligned} \quad (3.5)$$

Onde p_i são componentes da matriz P e ω_{ij} os componentes da matriz Ω . O modelo alternativo, como descrito anteriormente, difere do modelo de referência pela simplificação de μ para uma estimação pontual, igualando-o ao prêmio de risco de equilíbrio Π . A

consequência primária desta simplificação é a perda de significado da equação (3.2) – pois $\hat{\Pi}$ torna-se também uma estimação pontual – seguida da mudança no significado da variável τ . Enquanto no modelo de referência τ é relacionada à incerteza do CAPM e do resultado da otimização reversa, no modelo alternativo τ torna-se uma variável de controle da combinação entre o prêmio de equilíbrio e as visões do investidor, e sua função confunde-se com a da matriz Ω . Dado este conflito, no modelo alternativo pode-se igualar τ à 1 e controlar a combinação de retornos apenas através de Ω .

Utilizando o modelo alternativo, Idzorek (2007) propõe um método intuitivo para a estimativa de Ω , onde o investidor estabelece um nível de confiança entre 0% e 100% para cada visão.

Em Walters (2014), o autor observa que este método pode ser simplificado adicionando uma variável de encolhimento α à equação (3.5). Assim, nesta abordagem com $\tau = 1$, Ω pode ser interpretado como uma variável de encolhimento, que passa a ser controlado por α através da equação reescrita (3.6), com C sendo o nível de confiança entre 0% e 100%.

$$\begin{aligned}\Omega &= \alpha \, diag(P\Sigma P') \\ \alpha &= \frac{1 - C}{C}\end{aligned}\tag{3.6}$$

4 REDES NEURAIS

4.1 Introdução: Um breve histórico da Inteligência Artificial

O surgimento muito recente da área de Inteligência Artificial (IA) foi um resultado natural da busca humana pelo entendimento sobre como pensamos e agimos. Esta busca, talvez tão antiga quanto a experiência consciente, encontrou inspirações nos desenvolvimentos da computação e da neurociência após a segunda guerra mundial, onde a Inteligência Artificial passou a empregar estas descobertas na busca da construção de entidades inteligentes.

As primeiras sugestões de IA podem ser encontradas nos anos seguintes à segunda guerra, como o influente trabalho de TURING (1950) e a proposta do teste de Turing. O teste de Turing foi uma proposta para identificar o nível de inteligência de uma máquina. Nele, caso um entrevistador humano não consiga distinguir se o objeto entrevistado é uma máquina ou outro ser humano, a máquina pode ser considerada inteligente. Certamente esta declaração deixa margem para muita discussão, mas esta proposta formalizou um conceito de inteligência que passou a ser utilizado como bússola para a construção de sistemas inteligentes.

De acordo com Russell e Norvig (2016), para que uma máquina seja considerada aprovada no teste, é necessário que possua as seguintes competências:

Processamento de linguagem: Para comunicação.

Representação de conhecimento: Para armazenagem do que ela aprende ou ouve.

Raciocínio automatizado: Para usar a informação armazenada para responder perguntas e encontrar novas conclusões.

Aprendizagem de máquina: Para se adaptar às novas circunstâncias e extrapolar padrões de informações.

O teste original evita intencionalmente o contato físico entre entrevistado e entrevistador. Porém, o chamado teste de Turing total, uma versão ainda mais sofisticada, inclui um sinal de vídeo e uma escotilha, para intercâmbio de objetos e interação visual. Neste caso, pode-se acrescentar ainda outras duas competências:

Visão Computacional: Para distinguir objetos e interações visuais.

Robótica: Para movimentação e manipulação de objetos.

Estas seis áreas compreendem grande parte das pesquisas de Inteligência Artificial, e neste contexto o teste de Turing ainda é relevante, mesmo após sete décadas.

Desde sua concepção, a área de IA vem aumentando de maneira natural, com períodos de crescimento alternados com períodos de desinteresse. Conforme a capacidade de computação torna-se suficiente para as aplicações de Aprendizagem de Máquina – junto a algoritmos cada vez mais eficientes – as maiores limitações encontram-se na quantidade de dados disponíveis para treinamento. Porém, conforme a indústria e a sociedade avançam para a era do *Big Data* e 'Internet das Coisas', a quantidade de informação disponível tende a continuar crescendo. Diante deste novo paradigma, a capacidade humana de processamento de dados torna-se limitada quando comparada a estes sistemas automatizados, conferindo um poder de disrupção muito grande a estas tecnologias. Assim, estes sistemas trazem um potencial de remodelar toda a sociedade e as formas de trabalho, no que está sendo chamado de quarta revolução industrial. Grande parte da responsabilidade dos resultados obtidos pela aprendizagem de máquina deve-se às redes neurais artificiais, onde o bloco fundamental de construção é o neurônio artificial.

4.2 Neurônio Artificial

Tal qual sua inspiração biológica, a unidade básica da rede neural artificial é o neurônio. O neurônio artificial é, em sua essência, um interpretador de sinais. A topologia deste interpretador é composta de uma coleção de entradas, com bioinspiração nos dendritos, uma central de processamento de sinais, e apenas uma saída, inspirada no axônio do neurônio biológico. O núcleo do neurônio, a central de processamento de sinais, utiliza o sinal de entrada para calcular o sinal de saída através de uma função de ativação. A figura 1 mostra um diagrama simplificado do neurônio artificial.

Como abstração matemática, o neurônio artificial é considerado um agente linear. A saída do neurônio – y_s na figura 1 – pode ser entendida como a saída de uma função de ativação – Φ_s – para a somatória de entradas ponderadas pelos seus pesos sinápticos –

Figura 1: Neurônio artificial

$$\sum w_i x_i.$$

A natureza das redes neurais, e outras abordagens de aprendizagem de máquina, é generalizar a solução de problemas tal qual a mente humana. Diferente dos sistemas especialistas – onde o programador busca a relação de ‘todos os problemas conhecidos’ com ‘todas as soluções conhecidas’, sugerida por um especialista – os sistemas de aprendizagem de máquina buscam a maleabilidade na relação solução-problema inspirada na plasticidade neural. Isto significa que as redes neurais podem possuir a mesma arquitetura para resolver problemas de diferentes naturezas, e a chave desta maleabilidade é o processo de treinamento e aprendizagem da rede. Observando novamente a figura 1, a plasticidade do neurônio artificial reside nos pesos sinápticos (w_i). Através do processo de treinamento e aprendizagem, descritos adiante, busca-se os valores de pesos sinápticos que produzem o melhor resultado, para um problema específico, na saída do neurônio.

Como agente matemático o neurônio artificial é bastante limitado. Apenas um neurônio não é o suficiente para a generalização de solução de problemas. O estudo do conjunto de neurônios artificiais organizados em redes deu início ao ramo do conexionismo, dentro das faculdades da aprendizagem de máquina.

O perceptron, sugerido em Rosenblatt (1958), foi um marco no conexionismo por apresentar uma técnica prática para treinamento de redes neurais. A partir do método de treinamento do Perceptron, Rosenblatt derivou o teorema de convergência do Perceptron, que afirma que para qualquer conjunto de dados linearmente separável, a método de treinamento garante a convergência a uma solução após um número finito de iterações. Uma função é dita linearmente separável se ela pode ser representada como a somatória de funções lógicas lineares. Como é comum em sistemas de aprendizagem de máquina, inicialmente a capacidade da rede perceptron foi superestimada. O desânimo que seguiu

à constatação dos limites do perceptron perdurou ao longo da década de 70 e início de 80, com o interesse sendo renovado a partir de progressos na construção de computadores melhores e mais baratos e novos algoritmos de treinamento, ocorrendo em meados dos anos 80 e início dos anos 90.

4.3 Conexionismo e Arquiteturas de Redes Neurais

O conexionismo é uma abordagem para modelagem matemática da cognição humana. Sua origem remonta à década de 40 e ao trabalho de McCulloch e Pitts (1943), que apresenta o que é considerado como a primeira construção lógica de um neurônio artificial e a concepção de redes de neurônios artificiais. Redes conexionistas, também chamadas de redes de neurônios artificiais – doravante redes neurais – são compostas de neurônios artificiais interconectados em diferentes configurações, ou arquiteturas. O perceptron de Rosenblatt, que marcou o fim de uma era no conexionismo com as limitações do modelo reconhecidas no livro de Minsky e Papert (1969), é a arquitetura básica composta por uma camada de neurônios artificiais. Como demonstrado no livro, a rede de perceptrons de uma camada não tem a capacidade de resolver problemas lógicos de separabilidade não linear, como o problema de lógica XOR (*exclusive or*). Os críticos das redes neurais consideraram que as limitações de processamento do perceptron seriam refletidas em redes mais sofisticadas, limitando as capacidades de solução de problemas cognitivos de alto nível – como o reconhecimento de padrões – que atraíam o maior interesse e investimento da comunidade científica de inteligência artificial do período. Minsky e Papert previram que redes de perceptrons com camadas intermediárias, conhecidas como camadas escondidas, teriam a capacidade de solução destes problemas, porém seriam impossíveis de se treinar. Esta limitação diminuiu o interesse e afastou investimentos na década seguinte, quando os sistemas especialistas foram foco de maior atenção.

A opinião científica mudou com o algoritmo da regra de delta generalizada de Rumelhart, Hinton e Williams (1986), para treinamento de redes com camadas escondidas. O algoritmo, conhecido também como *back-propagation*, e a revolução da computação na década de 80 renovou o interesse nas redes neurais. O perceptron com uma camada, que pode ser treinado pela regra de delta simples, pode ser observado na figura 2. A única camada desta arquitetura teoricamente pode conter quantos neurônios deseja-se colocar. Para um treinamento supervisionado utilizando a regra de delta simples, alimenta-se a rede simultaneamente com pares conhecidos de entrada-saída e os pesos das conexões são ajustados de modo a criar um padrão sináptico que melhor generalize a função que

Figura 2: Single-Layer Perceptron

deseja-se aproximar. Esta categoria de arquiteturas é conhecida como '*Feed-Forward*', pois o sinal de entrada é passado sempre para a camada adiante, até ser processado em um sinal colhido nos neurônios da camada de saída. Parte da responsabilidade da revolução conexionista da década de 80 e 90 deve-se ao livro de Rumelhart et al. (1987), e dos trabalhos do *Parallel Distributed Processing (PDP) group* do MIT, com modelos e discussões para sistemas de reconhecimento de fala, memória, raciocínio espacial entre outros. Este trabalho motivou uma série de outras arquiteturas, com destaque à categoria das redes recorrentes (*Recurrent Neural Networks (RNN)*). As RNNs possuem uma arquitetura onde parte do conjunto de informações de saída é realimentado à rede, produzindo uma memória de curta duração. As redes recorrentes trouxeram grandes avanços em sistemas de reconhecimento de fala, com trabalhos pioneiros como Elman (1990) e Jordan (1997).

4.3.1 Treinamento

Ao contrário do paradigma de programação por código estruturado, os programas de aprendizagem de máquina buscam construir agentes que aprendem a partir do estudo de suas próprias experiências. A partir desta perspectiva, visualiza-se a importância do processo de treinamento na construção e utilização de sistemas de redes neurais, e torna-se essencial sintetizar um paralelo entre suas modalidades de treinamento: aprendizagem não supervisionada, supervisionada e por reforço. A abordagem de treinamento é intrinsecamente relacionada à arquitetura da rede utilizada, e consequentemente ao problema que deseja-se estudar.

No processo de aprendizagem não supervisionada, a única informação fornecida para o algoritmo de aprendizagem são as entradas da rede. Grande parte das aplicações de redes

não supervisionadas relaciona-se à categorização (*clustering*) e separação de informações. Um exemplo de abordagem de aprendizagem não supervisionada é a aprendizagem por competição. Nesta abordagem, os neurônios competem entre si pela melhor resposta em relação a uma informação de entrada, e ao longo deste processo eles se tornam especializados em um padrão específico presente naquele conjunto de informação. Este é um dos princípios dos mapas auto-organizáveis, ou mapas de Kohonen Kohonen (1990).

O conceito de aprendizagem supervisionada relaciona-se à "aprender com um professor". Ou seja, espera-se que após uma etapa onde são apresentados exemplos de entrada e saída cuja relação é conhecida – ou rotulada – o agente passa a generalizar a relação para informações não rotuladas. Torna-se então essencial o papel do erro neste processo. O objetivo da etapa de aprendizagem supervisionada é minimizar o erro entre a saída produzida pela rede e a saída conhecida, rotulada, utilizando retroalimentação (*feedback*). A regra de delta e a técnica de *back-propagation*, apresentadas adiante, são exemplos desta aplicação. Espera-se assim que o agente mapeie uma função, ou comportamento, que relacione o estímulo do ambiente (entradas) e uma ação ou decisão (saída).

A aprendizagem por reforço pode ser entendida como uma abordagem que utiliza conceitos das outras duas categorias. Ao invés de se apresentar exemplos de relações entrada-saída, o comportamento da rede é premiado ou punido de acordo com a saída fornecida. Diferentemente do professor da aprendizagem supervisionada, a aprendizagem por reforço necessita de um 'crítico'. Esta abordagem também utiliza *feedback*, e busca-se otimizar uma função utilidade. Por este motivo, o algoritmo *back-propagation* também pode ser aplicado em sistemas de aprendizagem por reforço. O sistema *AlphaGo* da DeepMind, que venceu o primeiro colocado do ranking mundial do jogo 'Go' em 2017, Ke Jie, é um exemplo de rede treinada utilizando aprendizagem por reforço Silver et al. (2017).

Outra diferenciação dentro das abordagens supervisionada e por reforço é o conceito de *batch learning* (por lote) e *on-line learning*. Na abordagem por lote, o ajuste dos pesos sinápticos ocorre após a apresentação de todas as amostras de treinamento de uma determinada *epoch* (época) enquanto que na abordagem on-line o ajuste nos pesos é feito exemplo-por-exemplo. Ao ajustar os pesos sinápticos após um lote, a maior vantagem é que o ruído no sinal de erro acaba sendo parcialmente contornado e o valor ajustado pelo gradiente torna-se mais preciso. Porém, o método de treinamento por lote exige armazenagem da informação, o que nem sempre é viável. A escolha do método de treinamento é dependente do problema analisado.

4.3.2 A regra do Perceptron (Regra de Delta Simples)

A regra de delta, regra de aprendizado do perceptron, ou regra de Widrow-Hoff, foi derivada à partir do princípio de Hebb, que constata que 'neurônios que disparam juntos, permanecem juntos' e busca otimizar uma função utilidade pelo método do gradiente. Esta abordagem é utilizada tanto no aprendizado supervisionado quanto no aprendizado por reforço. Na aplicação de aprendizado supervisionado procura-se mapear a relação de entradas-saídas conhecidas aumentando ou diminuindo os valores dos pesos sinápticos, de acordo com o erro quadrático entre a saída produzida pela rede e a saída conhecida. Por este motivo este algoritmo também é conhecido por método dos mínimos quadrados (*Least Mean Squared (LMS) method*). Definindo-se os pesos iniciais de uma rede aleatoriamente, a atualização de pesos sinápticos (ω_{ij}) entre o neurônio i e j é dada por:

$$\begin{aligned}\omega'_{ij} &= \omega_{ij} + \Delta\omega_{ij} \\ \Delta\omega_{ij} &= -\alpha \frac{\partial \epsilon}{\partial \omega_{ij}} \\ \epsilon &= y_j - \Phi_j\left(\sum_i x_i \omega_{ij}\right)\end{aligned}\tag{4.1}$$

Onde ϵ é o erro entre a saída conhecida y_j e a saída da rede $\hat{y}_j = \Phi_j(\sum_i x_i \omega_{ij})$. A variável α representa a taxa de aprendizagem. Uma taxa de aprendizagem maior reflete um ajuste mais brusco de pesos sinápticos em direção ao gradiente do erro, uma taxa de aprendizagem menor reflete num ajuste mais suave.

A construção do Perceptron Multi-Camadas (*Multilayer Perceptron (MLP)*), como comentado anteriormente, tornou-se possível com a técnica de treinamento por *back-propagation*, e tal foi o impacto desta abordagem que o MLP passou a ser conhecido como a rede neural '*vanilla*' (baunilha), no sentido de ser a rede mais amplamente difundida.

4.4 Feed-Forward Multi-Layer Perceptron

O MLP tornou-se uma arquitetura bastante utilizada em sistemas adaptativos, na predição de séries temporais e em sistemas de trading ao longo da década de 90. Sua estrutura é composta basicamente por sequências de camadas de neurônios artificiais interconectados, como na figura 3, apresentando no mínimo uma camada de entrada, uma camada intermediária chamada de camada escondida – podendo ter qualquer número finito de camadas escondidas –, e uma camada de saída. Superando o problema de

separabilidade não-linear do perceptron, o MLP torna-se uma ferramenta poderosa para aproximação de funções não lineares, onde sua maior limitação encontra-se especialmente na capacidade de processamento dos computadores de uma geração, ou na quantidade de informações disponíveis para o seu treinamento. Por este motivo, o MLP é um dos grandes protagonistas da revolução da inteligência artificial observada na contemporaneidade.

Intuitivamente pode-se argumentar que a adição de camadas escondidas melhoraria a funcionalidade da rede como aproximador universal. Porém, Cybenko (1989) mostrou que MLPs com apenas uma camada escondida e um número finito de neurônios na camada escondida podem aproximar qualquer função contínua em subconjuntos compactos de R^n .

Figura 3: Multi-Layer Perceptron

O que observa-se na prática, portanto, é que camadas escondidas adicionais não estão necessariamente relacionadas ao melhor desempenho na aproximação de funções contínuas. Como viés, o emprego de camadas adicionais pode aumentar drasticamente o tempo de treinamento sem melhoria na performance, ou aumentar o risco de *overfitting*, prejudicando a capacidade de generalização da rede neural.

Como em modelos estatísticos, o sobre-treinamento da rede neural pode refletir em *Overfitting* ou sobreajuste. O modelo com *overfitting* se ajusta demasiadamente aos dados de treinamento, perdendo a capacidade de generalização e incorrendo em baixo desempenho quando utilizado em outras amostras de dados. Ou seja, no caso de construções de redes com muitos nós e camadas, ou com amostras de treinamento pequenas, as redes podem 'memorizar' os dados e produzir baixos erros em relação às saídas conhecidas. Estas redes não necessariamente serão úteis em aplicações práticas.

Como comentado anteriormente, o que limitou a construção de uma rede Perceptron com camadas intermediárias foi a falta de um algoritmo para treinamento, visto que a regra de delta não 'enxerga' as camadas escondidas. A regra de delta generalizada,

ou o algoritmo de *back-propagation* de Rumelhart, Hinton e Williams (1986), trouxe a abordagem necessária para este avanço.

4.4.1 O Algoritmo Back-Propagation

Observando a figura 4, o modelo matemático do neurônio artificial s da camada j , sendo i a quantidade de sinais vindo da sua esquerda, pode ser representado pelas equações:

$$\begin{aligned} u_s &= \sum^i w_{ij}x_i \\ y_s &= \phi(u_s + b_s) \end{aligned} \quad (4.2)$$

Esta versão expandida da figura 1 apresenta como diferença a saída do somador (u_s) –

Figura 4: Neurônio Artificial (Expandido)

conhecida por campo local induzido (*Induced Local Field*) – e o viés (*bias*) b_s . Diferente da representação anterior, aqui foi incluído o componente de viés. O viés é responsável por mudar a resposta da função de ativação ao longo do eixo das abcissas, caso esta alteração seja necessária ao longo do processo de treinamento da rede. Ele pode ser entendido como um sinal fixo de valor unitário relacionado a um peso de viés para cada neurônio. Assim, um MLP composto por camadas escondidas sequenciais finitas, com um neurônio na camada de saída, pode ser descrito matematicamente pela expansão da equação (4.2) – onde as saídas da camada à esquerda tornam-se as entradas da camada seguinte – obtendo-se a equação (4.3).

$$\begin{aligned}
\hat{y}_s &= \phi_s\left(\sum_s w_{rs}y_r\right), \quad y_r = \phi_r\left(\sum_r w_{qr}y_q\right), \quad y_q = \phi_q\left(\sum_q \dots\right) \\
\hat{y}_s &= \phi_s\left(\sum_s w_{rs}\phi_r\left(\sum_r w_{qr}\phi_q\left(\sum_q \dots\phi\left(\sum_i w_ix_i\right)\right)\right)\right)
\end{aligned} \tag{4.3}$$

O método de *back-propagation* de aprendizado supervisionado, ou seja, aquele onde são analisados pares de saídas conhecidas e saídas produzidas pela rede, pode ser entendido em duas etapas. Na primeira etapa – *feed-forward* – o sinal de entrada propaga-se da camada de entrada adiante pelas camadas da rede até a camada de saída, produzindo o sinal de saída estimado $\hat{y}_s$. Na segunda etapa – *backpropagation* – calcula-se o sinal de erro ($y_s - \hat{y}_s$), utilizando as saídas conhecidas y_s , e este propaga-se retrogradamente ajustando os pesos sinápticos.

Assim, desde que todas as funções de ativação sejam diferenciáveis no domínio analisado, pode-se aplicar o método do gradiente – equação (4.1) – para o cálculo do conjunto de pesos que aproxime a função analisada.

4.4.2 Funções de Ativação

Para aplicar o método do gradiente na equação (4.3), portanto, as funções de ativação devem ser diferenciáveis no intervalo analisado. Um grande número de funções de ativação são utilizadas hoje, sendo a sua escolha normalmente feita de maneira empírica e dependente do problema analisado.

A primeira função sugerida em McCulloch e Pitts (1943) é a função degrau. Os autores sugerem a lógica *all-or-none* (tudo ou nada) a partir da bioinspiração do neurônio real. Ou seja, caso o neurônio atinja uma excitação acima de um gatilho, o neurônio torna-se ativo na rede, causando um efeito em cadeia. O perceptron de Rosenblatt utiliza a função degrau, implementando a lógica de McCulloch e Pitts. Porém, a grande limitação da rede de perceptrons é que a função degrau não é diferenciável no ponto 0, o que torna inviável a sua utilização em redes multi-camadas. A figura 5 apresenta o gráfico da função degrau.

A alternativa mais óbvia para uma função contínua e diferenciável é a função linear, $\Phi(\nu) = \nu$, figura 6, com alcance de valores de $-\infty$ à $+\infty$. Porém a função linear não ajuda na criação de redes de elevada complexidade, pois perde-se a capacidade de representação de funções não lineares. Além disso, a derivada da função é constante – o que resulta em não convergência no método do gradiente. Caso o MLP tenha funções lineares em todas

Figura 5: Função degrau

Figura 6: Função linear

as suas camadas a rede torna-se uma combinação de funções lineares, e pode-se mostrar utilizando álgebra linear que qualquer número de camadas pode ser reduzido à uma rede de duas camadas, entrada-saída. É comum a utilização de funções lineares apenas em camadas de saída de MLPs.

Uma função bastante utilizada após o surgimento do algoritmo de *back-propagation*, e utilizada ainda hoje em diferentes aplicações, é a função logística, ou sigmoide. A função logística apresenta o que é conhecido como *soft-thresholding*, ou passo suave, trazendo a função degrau para uma versão contínua e diferenciável. Seu comportamento assemelha-se a uma curva em forma de S, como pode ser observado na figura 7 e equação (4.4). O intervalo de atuação restringe-se entre os valores 0 e 1. A inclinação da função logística é um parâmetro ajustável no momento de criação da rede, através da variável a , onde é possível configurar a velocidade que o neurônio é ativado na rede, como mostra a figura 8.

$$\Phi(\nu) = \sigma(\nu) = \frac{1}{1 + e^{-a\nu}} \quad (4.4)$$

A tangente hiperbólica é uma alternativa à função logística. Seu comportamento é seme-

Figura 7: Função passo suave (Sigmoide)

Figura 8: Inclinação do Passo Suave

Figura 9: Tangente Hiperbólica

Figura 10: Derivadas da Tangente Hiperbólica e Sigmoide

lhante, alterando apenas sua escala, ficando definido entre o intervalo -1 e +1 como pode ser observado na equação (4.5) e figura 9. Outra vantagem em relação à função logística é a inclinação da sua derivada, melhorando o desempenho do método do gradiente.

$$\Phi(\nu) = \tanh(\nu) = \frac{e^\nu - e^{-\nu}}{e^\nu + e^{-\nu}} = \frac{2}{1 + e^{-2\nu}} - 1 \quad (4.5)$$

Nota-se pela figura 10 o comportamento das derivadas da função logística e tangente hiperbólica. Durante o processo de *back-propagation* o erro é representado em cadeias de derivadas das funções de transferência.

Para redes com muitas camadas, como é o caso das redes de *deep-learning*, parte da informação dos erros acabam se perdendo ao longo deste comportamento de baixa inclinação, resultando em não convergência. Por este motivo, as aplicações de redes neurais mais recentes que utilizam muitas camadas buscaram soluções em outra classe de funções de transferência que tentam resgatar o comportamento do neurônio biológico. É o caso da função *Rectified Linear Unit* (ReLU), definida pela equação (4.6) com comportamento observado na figura 11. Para valores abaixo do gatilho 0, a função retorna valor nulo, para valores acima do gatilho ela possui uma saída linear. À primeira vista seu comportamento causa estranheza por não ser diferenciável no ponto 0, o que leva a questionamentos quanto ao seu uso junto ao algoritmo de *back-propagation*. Porém, na prática, dificilmente a otimização chega ao ponto de zerar a função de erro, e esta descontinuidade de fato não se concretiza em um problema. O comportamento da derivada do ReLU assemelha-se à função degrau, descrito pela equação (4.7).

$$\Phi(\nu) = \operatorname{argmax}\{0, \nu\} \quad (4.6)$$

Figura 11: Função ReLU

$$\Phi'(\nu) = \begin{cases} 0, & \text{para } \nu < 0, \\ 1, & \text{c.c} \end{cases} \quad (4.7)$$

Outras funções que buscam a mesma representação deste comportamento são: *Exponential Linear Unit* (ELU), *Leaky ReLU* e *Parametric ReLU* (PrELU).

5 MODELO PROPOSTO E METODOLOGIA

O objetivo do modelo proposto é buscar evidências favoráveis à utilização da abordagem bayesiana em sistemas autônomos de investimento e gestão de patrimônio. Para este fim foi construído um sistema composto por dois núcleos. O primeiro núcleo é responsável pelo monitoramento do mercado utilizando redes neurais. O segundo núcleo trata da inferência bayesiana em si, através do modelo de Black-Litterman, utilizando as análises do primeiro núcleo para sugerir uma carteira de posições casadas de compra e venda para as ações monitoradas. Os resultados do modelo são então comparados aos de um segundo sistema autônomo de investimento, sem a utilização da inferência bayesiana, e as métricas de desempenho são analisadas sob uma ótica de risco e retorno.

O modelo foi implementado utilizando ferramentas *open-source*. Utilizou-se a API Keras, Chollet et al. (2015), com a *engine* TensorFlow para a construção das redes neurais e as bibliotecas SciPy, Jones et al. (2001) e NumPy, Oliphant (2006), para manipulação de dados.

Com o objetivo de diminuir as exigências computacionais, o sistema proposto monitora um mercado com número reduzido de ativos. Para tanto, foram escolhidas as dez ações com maior peso do índice Bovespa ao longo do período analisado, de 30/04/2014 à 29/04/2016. Os ativos escolhidos são: Ambev SA [ABEV3], Banco Bradesco SA [BBDC4], BRF S.A. [BRFS3], Cielo SA [CIEL3], Itaúsa Investimentos Itaú SA [ITSA4], Petrobrás ON [PETR3], Petrobrás PN [PETR4], Ultrapar Participações SA [UGPA3], e Vale [VALE3].

O sistema funciona sob as seguintes regras: As posições são sugerida ao final do dia, após o fechamento do mercado, para o pregão do dia seguinte. Utilizam-se como entradas a carteira Ibovespa do final do pregão para o cálculo dos retornos de equilíbrio, e as previsões de preço de fechamento para o d_{i+1} , feitas pelo conjunto de redes neurais do primeiro núcleo transformadas em previsões de retornos. No início do pregão seguinte montam-se as operações e ao final do pregão liquida-se toda a carteira. Então, novamente é feita a previsão de preços do dia seguinte, e as etapas se repetem. A figura 12 ilustra o

funcionamento diário do sistema de maneira simplificada.

Obviamente, uma atualização diária da carteira resulta em elevados custos operacionais. Como o objetivo deste sistema é analisar o modelo de Black-Litterman utilizado em conjunto com as previsões de um sistema quantitativo autônomo, os custos operacionais foram desconsiderados.

Figura 12: Diagrama Simplificado do Sistema

5.1 Primeiro Núcleo: Implantação das Redes Neurais

A função do primeiro núcleo do sistema é acompanhar e monitorar o mercado, e oferecer um conjunto de previsões de oscilações para o segundo núcleo. O primeiro núcleo é composto por um conjunto de dez redes neurais. As redes neurais foram construídas baseadas em resultados de trabalhos anteriores, e suas arquiteturas foram definidas por um processo seletivo utilizando busca exaustiva. Neste processo seletivo, treinou-se o maior número possível de redes neurais por ativo, utilizando-se um conjunto de dados, e foram selecionadas as arquiteturas de redes que frequentemente produzissem boa performance, quando utilizadas para prever oscilações em outro conjunto de dados. Um terceiro conjunto de dados – onde o modelo proposto foi efetivamente testado – foi apresentado apenas para as arquiteturas vencedoras, e apenas após a etapa de retreinamento utilizando janelamento móvel. A base de dados consiste em históricos de preços diários das ações analisadas, trabalhados utilizando indicadores de análise técnica. A figura 13 ilustra as etapas para implantação de uma rede neural do primeiro núcleo.

Buscou-se implementar modelos de redes neurais com bons resultados em trabalhos anteriores. Seguindo orientações de Kaastra e Boyd (1996), Klassen (2007) e Gambogi e

Figura 13: Etapas de implantação do primeiro núcleo

Costa (2013), as redes implementadas são variações do *Feed-Forward Multi-Layer Perceptron*, com uma camada escondida. Todas as redes foram treinadas utilizando treinamento supervisionado com a técnica de *cross-validation* para diminuir o problema de *overfitting*. Utilizou-se a função de transferência tangente hiperbólica nas camadas escondidas, com funções lineares na camada de saída. O universo de possíveis entradas e arquiteturas foi reduzido a partir de busca exaustiva, até restar uma arquitetura vencedora por ativo, como descrito adiante.

5.1.1 Indicadores da Análise Técnica

Para compor o universo de entradas, foram escolhidos os indicadores de análise técnica utilizados em Klassen (2007), com as modificações sugeridas por Gambogi e Costa (2013). São eles: MACD (*Mean Average Convergente Divergence*) com períodos de 8 e 17 dias, ROC (*Rate of Change*) com período de 10 dias, além dos preços de fechamento do dia e dos últimos três dias ($d_i, d_{i-1}, d_{i-2}, d_{i-3}$). As redes devem fazer a previsão de um dia no futuro, portanto cada rede possui uma única saída, correspondente à previsão de preço em (d_{i+1}).

O indicador MACD, descrito em Murphy (1999), busca captar modificações de momentum e direção de variações de preços. Ele é obtido através da diferença de uma média móvel 'rápida' e uma média móvel 'lenta'. A interpretação entre os analistas técnicos desta diferença é que os preços tendem a se manter próximos à média lenta, e consequentemente

esta diferença constitui uma oportunidade de arbitragem. Embora esta interpretação seja irrelevante para a rede neural, ao aplicá-la espera-se que de alguma maneira este comportamento esteja refletido em preços futuros. A equação (5.1) representa o cálculo do indicador. O indicador ROC, equação (5.2), também descrito em Murphy (1999) e acrescentado em Gambogi e Costa (2013), é um indicador puro de momentum e registra a variação de preço ao longo de um período.

$$MACD = MME(p_1) - MME(p_2) \quad (5.1)$$

$$ROC(p) = \frac{d_i - d_{i-p}}{d_{i-p}} \quad (5.2)$$

onde $MME(p)$ é a Média Móvel com decaimento Exponencial de período p , que pode ser implementada por:

$$MME(p) = \frac{P_p + (1 + \alpha)P_{p-1} + (1 + \alpha)^2 P_{p-2} + \dots + (1 + \alpha)^p P_0}{1 + (1 + \alpha) + (1 + \alpha)^2 + \dots + (1 + \alpha)^p}$$

$$\alpha = \frac{2}{p + 1}$$

Com P_i sendo o preço de fechamento no dia i .

5.1.2 Conjuntos de Treinamento, Teste e Verificação

O conjunto total de dados utilizado foi dividido em três subconjuntos mutuamente exclusivos: treinamento, teste e verificação. A amostra de treinamento utiliza também a abordagem de treinamento/validação, que é outra sugestão para diminuição de *overfitting* e é nativa de algumas APIs, como o Keras. O modelo busca minimizar o erro na amostra de treinamento enquanto valida esta minimização em paralelo em uma amostra de validação. O processo de otimização é interrompido no ponto onde ocorre divergência entre os gradientes de erro de treinamento e validação. O primeiro e segundo conjunto, treinamento/validação e teste, pertencem ao período de 30/04/2011 à 29/04/2014. Estes dados foram embaralhados, sendo que durante o treinamento as redes 'não sabiam' se tratar de uma série contínua. Após o embaralhamento, dividiu-se os dados efetivamente em treinamento/validação e teste em uma proporção de 80% e 20%. Após a finalização do treinamento as redes foram expostas ao conjunto de dados de teste para prever os preços em d_{i+1} . Utilizando estas previsões, foram selecionadas as arquiteturas que frequente-

mente produziram boas métricas de desempenho, segundo os critérios apresentados nas próximas seções.

O terceiro conjunto, verificação, correspondeu ao período de 30/04/2014 à 29/04/2016. O conjunto de verificação foi utilizado apenas com as redes vencedoras, e neste período o sistema foi efetivamente testado utilizando os dois núcleos.

As entradas das redes neurais foram normalizadas entre o intervalo de -0.75 à 0.75, como sugerido nos trabalhos relacionados, para estimular as redes a trabalhar dentro da região linear da função de transferência, minimizando os efeitos de valores muito grandes ou muito pequenos no cálculo do erro de *back-propagation*.

5.1.3 Seleção de Arquiteturas

A estratégia de maximizar as rodadas de treinamentos utilizando loops foi escolhida como opção para lidar com o problema dos mínimos locais utilizando busca exaustiva. O universo de arquiteturas possíveis foi reduzida utilizando abordagens de trabalhos anteriores. Assim, a filtragem partiu de dez arquiteturas definidas *a priori*, descritas na tabela 1.

	d_i	d_{i-1}	d_{i-2}	d_{i-3}	MACD	ROC	<i>hidden</i>
I	●	●	●	●	●	○	3
II	●	●	●	●	●	○	2
III	●	●	●	●	○	●	3
IV	●	●	●	●	○	●	2
V	●	●	●	○	●	○	2
VI	●	●	●	○	○	●	2
VII	●	●	○	○	●	○	1
VIII	●	●	○	○	○	●	1
IX	●	○	○	○	●	●	2
X	●	○	○	○	●	●	1

Tabela 1: Arquiteturas escolhidas *a priori*

Cada arquitetura possui o número de entradas variáveis, o número de camadas escondidas definidos pela variável *hidden* e uma camada de saída. As camadas escondidas possuem função de transferência tangente hiperbólica e as camadas de saída possuem função de transferência linear. Para cada uma destas arquiteturas foram treinadas quinhentas redes. As redes foram aleatoriamente separadas em cinco séries de cem, e estas cinco séries foram paralelamente expostas à filtragem, sendo escolhida uma rede vitoriosa por série, como ilustra a figura 14. Ou seja, cinco redes vencedoras por arquitetura.

Figura 14: Filtragem por série de treinamentos

Como critério de filtragem levou-se em consideração três atributos de performance: O retorno excedente produzido pela rede, em comparação à estratégia de *buy and hold* no período de teste; A taxa de acertos na previsão de operações vencedoras no período (*score*), onde uma operação é contabilizada como vencedora quando a oscilação de preço do ativo ocorre na direção prevista pela rede, independente do preço previsto, e contabilizada como perdedora no caso contrário; e a raiz do erro médio quadrático (*RMSE: Root Mean Squared Error*) das previsões de preços.

A métrica de retornos excedentes produzido pela rede foi calculada como uma simulação da rede sendo utilizada automaticamente em uma estratégia de *trading*. Nesta simulação, a rede prevê o preço futuro em d_{i+1} e assume a posição comprada em caso de previsão de alta, ou assume posição vendida em caso de previsão de baixa. A métrica resultante é então o retorno excedente produzido, descontado do retorno do mesmo ativo no período. Optou-se por utilizar o retorno excedente como indicador sobre o retorno absoluto da rede, pois os dados que compunham o conjunto de teste eram variáveis entre as séries, devido ao embaralhamento. O *score* é calculado de maneira semelhante, contabilizando as operações vencedoras da rede. Embora o RMSE seja efetivamente a função que procurou-se minimizar durante o treinamento, a boa performance do sistema está relacionada à previsão correta de oscilações, motivo pelo o qual o retorno excedente no período e o *score* da rede foram escolhidos como principais atributos de filtragem. O RMSE foi considerado como critério de desempate.

Após a aplicação do filtro, o universo de redes foi reduzido à cinco redes vencedoras por arquitetura. Calculou-se então a média das métricas (Retornos, Score, RMSE) destas redes e aplicou-se novamente a metodologia do filtro. A arquitetura vencedora por ativo, como ilustrado pela figura 15, foi aquela com melhor taxa de acerto médio entre as três arquiteturas de melhor retorno excedente médio, sendo o RMSE novamente utilizado como critério de desempate.

Figura 15: Filtragem de arquiteturas

Ativo	Arquitetura
ABEV3	VI
BBDC4	VII
BRFS3	X
CIEL3	VIII
ITSA4	V
ITUB4	X
PETR3	VII
PETR4	I
UGPA3	I
VALE3	III
IBOV	III

Tabela 2: Arquiteturas Vencedoras

O objetivo desta metodologia foi selecionar a arquitetura que resultasse em bons resultados com mais frequência, com os resultados descritos pela tabela 2, para posteriormente ser aperfeiçoada na etapa de janelamento móvel.

Ao longo deste processo, como pode ser observado na tabela 2, também foi treinada uma rede para o índice Bovespa [IBOV]. Esta rede foi posteriormente utilizada na construção do *Benchmark*, ilustrado mais adiante.

5.1.4 Janelamento Móvel

Idealmente, poderia-se utilizar o treinamento *on-line* das redes neurais conforme novas informações diárias fossem fornecidas. Devido à exigência computacional desta aborda-

gem, optou-se pela alternativa de treinamento por lotes utilizando janelamento móvel.

A abordagem de janelamento móvel busca retreinar a rede neural a determinados intervalos. Para isto, divide-se a amostra total de dados (treinamento, teste e verificação) pela quantidade de janelas desejadas, a cada início de período de verificação, utiliza-se o período imediatamente anterior para treinamento e testes, como ilustra a figura 16.

Figura 16: Janelamento

O embasamento desta abordagem encontra-se no caráter dinâmico e adaptativo do mercado financeiro, conhecido intuitivamente pelos operadores do mercado e presente em algumas formulações epistemológicas da formação de preços, como a hipótese do mercado adaptativo, Lo (2004) e a hipótese do mercado fractal Peters, Peters e Peters (1994).

Na metodologia utilizada neste trabalho, após a definição das arquiteturas vencedoras por ativo, passou-se à etapa de treinamento por janelamento móvel.

5.1.4.1 Tamanho das Janelas

Ainda assim, dentro da abordagem de treinamento por janelamento, poderia-se escolher tamanhos de janelas ideais para cada ativo analisado, o que necessitaria de uma nova etapa de otimização que escolheria o tamanho ideal de janela por ativo. Como feito anteriormente, na limitação de arquiteturas *a priori* e na escolha de funções de transferência, optou-se por uma abordagem de padronização para limitar o uso de recursos computacionais. Assim sendo, optou-se pela utilização de tamanhos de janelas comum a todos os ativos.

Note na tabela 2 que foi escolhida uma arquitetura para o índice IBOV. Embora o Ibovespa não seja efetivamente um ativo negociado em bolsa, sua variação de pontos foi tratada como um ativo financeiro nesta etapa, para o dimensionamento de um tamanho único de janelas. A rede neural selecionada para compor o sistema de *Benchmark*, como será explicado adiante, também utilizou esta mesma arquitetura.

Os possíveis tamanhos de janelas foram definidos a partir do subconjunto de verificação. Este subconjunto foi composto por 494 amostras. Foram testadas janelas com

dimensões $-(J)-$ de 13, 19, 26 e 38 amostras, múltiplos de 494. O tamanho das janelas de treinamento $-(JT)-$ deveriam ser números inteiros que maximizassem a utilização das amostras. O conjunto total de treinamento e teste era composto de 743 amostras, então a dimensão T foi definida pelo valor inteiro da divisão, como segue:

$$T = \left\lfloor \frac{743}{J} \right\rfloor$$

Para cada combinação de configuração de janela $J - T$ – utilizando-se apenas a amostra de treinamento e teste – aplicou-se novamente a metodologia de filtragem. Seguindo metodologia semelhante à etapa de seleção de arquitetura, foram treinadas 100 redes por janela, para cada configuração. Estas 100 redes foram separadas aleatoriamente em 5 séries de 20, das quais foram selecionadas 5 redes vencedoras por janela, utilizando as métricas de Retorno Excedente, Score e RMSE.

Janelas	Amostras de treinamento e teste	Amostras de verificação
38	741	13
26	741	19
19	728	26
13	722	38

Tabela 3: Configurações de Janelas

Assim, foram calculadas as médias destas 5 vencedoras por janela, resultando em um indicador de desempenho para cada janela. Naturalmente, a quantidade de janelas por configuração $J - T$ é variável, e conseqüentemente a quantidade de vencedoras por configuração é relacionada a esta grandeza, como mostra a tabela 3. Para poder comparar as diferentes configurações de janela, optou-se por tirar uma média destes indicadores, resultando na tabela 4. A configuração 38 – 19 obteve as melhores métricas médias.

J-T	Média de Retornos	Score Médio	RMSE Médio
13-57	8.71%	52.26%	1.23%
19-39	5.79%	51.47%	1.36%
26-28	6.31%	51.60%	1.33%
38-19	9.49%	52.33%	1.28%

Tabela 4: Métricas das Configurações de Janelas no período de Testes

5.1.5 Treinamento das Redes Neurais e Saída do Primeiro Núcleo

A última etapa do primeiro núcleo foi o treinamento das redes utilizando as dimensões de janelas definidas na etapa anterior. Como comentado anteriormente, isto significa um retreinamento contínuo a cada início de janela, como ilustra a figura 16.

Cada treinamento foi acompanhado de uma etapa de busca exaustiva de melhor conjunto de pesos. Para cada janela foram treinadas 100 redes neurais, separadas em 20 grupos de 5. Entre as 5 finalistas, foi escolhida uma rede vencedora. A figura 17 ilustra esta última etapa de filtragem. O produto desta etapa é um conjunto de pesos sinápticos por janela por ativo.

Figura 17: Treinamento de uma Janela

Como exemplo de resultado do primeiro núcleo, considere o ativo ABEV3. A arquitetura ótima encontrada foi a de número VI, repetida na tabela 5 para facilitar a visualização, e que também pode ser representada pelo diagrama da figura 18.

	d_i	d_{i-1}	d_{i-2}	d_{i-3}	MACD	ROC	<i>hidden</i>
VI	●	●	●	○	○	●	2

Tabela 5: Arquitetura ótima: ABEV3

Figura 18: ABEV3

Os dados da primeira janela de treinamento/teste, limitada pelo período de 31/05/2011 a 29/04/2014, foram divididos de forma que 80% dos dados foram utilizados para treinamento e os outros 20% foram separados. Após treinamento foram obtidos 100 conjuntos de pesos sinápticos na forma da tabela 6.

	$[h_0, h_1]$		$[o]$
i_0	[0.66642487, -0.86955923]	h_0	[0.9292472]
i_1	[0.27950108, -0.695755]	h_1	[-1.25893271]
i_2	[0.88129503, -0.58044165]	bias	[0.13739333]
i_3	[-0.55097717, -0.62097532]		
bias	[0.09154823, -0.05482146]		

Tabela 6: Tabela de pesos sinápticos para ABEV3 - Janela 1

Então, utilizando os dados de treinamento – 20% da amostra – alimentou-se estas 100 redes em paralelo, e analisou-se a saída obtida $\hat{y}_s$ em comparação com a saída conhecida y_s . Destas 100 redes, foi escolhida aquela com melhores métricas, cujos valores são estes da tabela 6. Esta rede foi utilizada então para previsões ao longo do período de 30-04-2014 a 25-06-2014. Note que as saídas desta rede se encontram em valores normalizados, que para uso do segundo núcleo precisaram passar pelo processo inverso de normalização. Estas saídas ficam então na forma de uma previsão de preço, que são transformadas em previsão de retorno utilizando os valores de fechamento do dia anterior. Seguindo o exemplo, para o caso da ABEV3 no dia 30-04-2014, a saída do primeiro núcleo – após processo inverso de normalização – foi R\$ 14,57 sendo o valor de fechamento do dia anterior R\$ 14,37, ou seja, a previsão de variação de preço calculada foi de:

$$\frac{14,57 - 14,37}{14,37} = 1,39\%$$

Ao final do período é efetuado um novo treinamento da rede utilizando os dados de 25-07-2011 a 25-06-2014, e os pesos sinápticos obtidos são utilizados no período seguinte de 26-06-2014 a 19-08-2014. Assim seguiu com todos os ativos ao longo de todas as janelas.

Esta etapa foi repetida para cada um dos dez ativos, além do índice IBOV para construção do sistema de *Benchmark*. A tabela 7 apresenta uma síntese do desempenho médio destas redes calculadas com dados anteriores a 29/04/2014. As informações desta tabela foram posteriormente utilizadas como confiabilidade das previsões no modelo de Black-Litterman.

5.2 Segundo Núcleo: Modelo de Black-Litterman

O objetivo do segundo núcleo do sistema é receber as informações do primeiro núcleo e, juntamente com as informações da carteira Ibovespa diária, propor uma nova carteira diária utilizando posições de compra e venda para os dez ativos analisados, que correspondem aos dez ativos mais negociados no período do experimento. Devido à simplificação de redução do universo de ativos utilizados, o núcleo do otimizador de carteiras utilizou a composição diária do Ibovespa ajustada – calculado utilizando a equação (5.3) – para calcular o retorno excedente de equilíbrio diário Π_i . Alternativamente, como em Xing, Cambria e Welsch (2018), o retorno de equilíbrio poderia ser calculado utilizando-se o método CAPM.

$$w_{i_{ajustado}} = \frac{w_i}{\sum_{10+} w_i} \quad (5.3)$$

O conjunto de previsões do primeiro núcleo, para cada dia, foi agrupado no vetor Q_i de visões dos investidores. Este vetor é composto pela saída das dez redes neurais transformadas em retornos esperados após processo inverso de normalização, como demonstrado ao final da seção 5.1.5.

Como estes retornos esperados são considerados não correlacionadas, a matriz P_i torna-se a matriz identidade. Foi utilizada uma matriz de covariância constante para todas os períodos do otimizador, calculada utilizando os dados dos conjuntos de treinamento e teste.

O retorno excedente de equilíbrio diário (Π_i) foi obtido através da otimização reversa

do índice ajustado, utilizando a equação (5.4). Baseando-se em trabalhos anteriores, adotou-se δ , o coeficiente médio de aversão ao risco da carteira, fixado em 2.5. A variável w_i é a carteira diária do índice Bovespa e Σ a matriz de covariância.

$$\Pi_i = \delta \Sigma w_{i_{ajustado}} \quad (5.4)$$

Utilizando a abordagem de Idzorek (2007), Walters (2014) e Satchell e Scowcroft (2000) para a implantação do modelo alternativo de Black-Litterman, foi definido $\tau = 1$ com o controle da combinação entre os retornos de equilíbrio e as estimativas das redes feita através da matriz Ω . Assim, para o cálculo dos retornos diários de Black-Litterman – com $\tau = 1$ e P_i igualando-se à matriz identidade – a equação (3.4) pode ser simplificada para a equação (5.5).

$$\hat{\Pi}_i = \Pi_i + \Sigma[\Sigma + \Omega]^{-1}[Q_i - \Pi_i] \quad (5.5)$$

5.2.1 Matriz Ω e confiabilidade α

Assim, a única variável ainda não definida é a matriz Ω . Como explicado anteriormente, na implantação do modelo alternativo pelo método de Idzorek, Ω é calculada utilizando uma abordagem de confiabilidade. Para as condições apresentadas, pode-se simplificar a equação (3.6) para a equação (5.6):

$$\begin{aligned} \Omega &= \alpha \text{diag}(\Sigma) \\ \alpha &= \frac{1 - C}{C} \end{aligned} \quad (5.6)$$

Onde Σ é a matriz de covariância da carteira – calculada utilizando valores históricos – e C é o vetor de confiabilidade composto pelo índice de acerto de operações das redes nos períodos de testes. O índice de acertos foi calculado considerando as operações vencedoras previstas pelas redes. Uma operação é considerada vencedora quando o retorno previsto pela rede ($\hat{y}_s$) possui o mesmo sinal do retorno real do mercado (y_s). Uma operação vencedora pode ser representada pela função passo de Heaviside, onde a operação j é vencedora quando $c_j = 1$ e perdedora se $c_j = 0$. Como descrito pela equação (5.7)

$$c_j = \frac{1 + \text{sgn}(s_j)}{2}$$

$$s_j = \text{sgn}(\hat{y}_s) * \text{sgn}(y_s)$$
(5.7)

A confiabilidade C_i da rede i pode então ser calculada contabilizando as operações vencedoras, ou pelo valor esperado de c_j , ou seja:

$$C_i = E[c_j] = \frac{\sum_j c_j}{j}$$
(5.8)

Os valores de confiabilidade das redes no período até 29/04/2014 estão expressos na tabela 7.

Ativo	Score
ABEV3	53.27%
BBDC4	51.30%
BRFS3	52.39%
CIEL3	51.08%
ITSA4	49.79%
ITUB4	51.63%
PETR3	50.35%
PETR4	51.81%
UGPA3	52.96%
VALE3	52.84%
IBOV	54.56%

Tabela 7: Confiabilidade das Redes

Como neste caso a variável α é um vetor, fica mais fácil analisar a equação (5.6) na forma da equação (3.5), ou seja:

$$\omega_{ij} = \alpha_i \Sigma_{ij}, \forall i = j$$

$$\omega_{ij} = 0, \forall i \neq j$$

onde :

$$\alpha_i = \frac{1 - C_i}{C_i}$$
(5.9)

Considere, por exemplo, o dia 02/05/2014. Os valores das variáveis do sistema para cada ativo podem ser observadas pela tabela 8.

A matriz Ω , tal qual a matriz Σ , são constantes nesta implementação. Utilizando Ω

	ABEV3	BBDC4	BRFS3	CIEL3	ITSA4	ITUB4	PETR3	PETR4	UGPA3	VALE3
$\hat{y}_s(\text{R\$})$	14,57	17,76	48,23	20,30	5,18	22,77	16,16	16,76	50,72	25,76
Q_i	1,40%	1,84%	0,04%	-0,79%	0,40%	-0,39%	3,55%	1,34%	-0,09%	0,12%
Π_i	15,39%	5,49%	11,91%	6,77%	16,40%	4,91%	12,63%	12,95%	12,08%	6,49%
$\hat{\Pi}_i$	1,52%	2,67%	1,17%	0,42%	2,45%	2,42%	4,63%	4,28%	0,84%	2,71%
w'_{i+1}	7,19%	8,45%	-1,17%	-3,27%	-3,36%	1,06%	5,41%	6,71%	-4,43%	1,90%

Tabela 8: Estados do sistema para o dia 30/04/2014

calculada a partir da tabela 7 e da equação (5.9), e substituindo os valores de Π_i e Q_i do dia 30/04/2014 na equação (5.5), chega-se aos retornos de Black-Litterman $\hat{\Pi}_i$, saída do segundo núcleo.

$$w'_i = (\delta\Sigma)^{-1}\hat{\Pi}_i \quad (5.10)$$

Os retornos de Black-Litterman, por sua vez, são utilizados então na equação de Markowitz sem restrições, equação (5.11), onde obtém-se a carteira w'_{i+1} para o dia 02/05/2014.

5.3 Sistema de Benchmark

O sistema de *Benchmark* foi construído utilizando as mesmas abordagens de construção do primeiro núcleo, tratando-se de uma implementação baseada no sistema de Gambogi e Costa (2013). Ao contrário do modelo proposto neste trabalho, este sistema de *Benchmark* monitora e opera apenas um ativo. A rede neural utilizada analisa as variações do índice Bovespa e se posiciona comprando ou vendendo o índice. Embora o índice não seja efetivamente um ativo negociado em bolsa, existem alguns ativos que seguem a variação do índice, como ETFs (*Exchange Trading Funds*).

A escolha do *Benchmark* pautou-se em criar um ambiente equivalente porém sem a utilização da inferência Bayesiana, representada pelo modelo de Black-Litterman. Após aplicação do mesmo processo seletivo do primeiro núcleo, a arquitetura III da tabela 1 foi a vencedora para o índice Bovespa. Utilizou-se também a técnica de janelamento móvel, com as mesmas dimensões de janela do sistema de dois núcleos.

O sistema de *Benchmark* utiliza as previsões de d_{i+1} fornecidas por uma rede neural, retreinada a cada início de janela de verificação, para se posicionar em relação ao índice Bovespa. Deste modo, quando a rede neural prevê uma oscilação positiva o investidor assume uma posição de compra do índice e quando a rede prevê uma oscilação negativa, assume-se uma posição de venda do índice. Ao final de cada dia de pregão, liquida-se a posição e repete-se a previsão do índice para o dia seguinte.

Tal qual no modelo de dois núcleos, o posicionamento do investidor é ponderado pela confiabilidade da rede neural durante o período de treinamento, com o Score médio da tabela 7, sendo que naquele esta confiabilidade alimenta o cálculo da matriz Ω , e neste o posicionamento é apenas ponderado de maneira linear.

Nestas condições, a valorização diária do patrimônio produzida pelo sistema de *Benchmark* é dada por:

$$V_{i+1} = s_i |p_{i+1} - p_i|$$

onde:

$$s_i = \begin{cases} +1 & \text{se } \text{senal}(p_{i+1} - p_i) = \text{senal}(y_{i+1} - p_i) \\ -1 & \text{se } \text{senal}(p_{i+1} - p_i) \neq \text{senal}(y_{i+1} - p_i) \end{cases}$$

onde p_i é o preço de fechamento no dia (d_i) , p_{i+1} o preço de fechamento em (d_{i+1}) , e y_{i+1} a previsão de preço de fechamento para (d_{i+1}) .

5.4 Otimização de Carteiras

Após processamento de saída das redes, com as previsões alimentando o modelo de Black Litterman e os retornos a posteriori obtidos, o processamento dos dados do segundo módulo pôde ser feito pelo método de otimização de carteiras. A abordagem recomendada para os retornos de Black-Litterman é a otimização de Markowitz sem restrições, para o cálculo de carteiras diárias. Na equação (5.11), a carteira w'_i representa a posição a ser tomada utilizando os retornos Π_i obtidos na saída do segundo núcleo, para cada dia i .

$$w'_i = (\delta \Sigma)^{-1} \hat{\Pi}_i \quad (5.11)$$

As saídas da otimização sem restrição são analisadas no capítulo seguinte.

5.5 Súmula

Em suma, o primeiro núcleo utiliza indicadores de análise técnica como entrada de redes neurais especializadas em cada ativo para produzir um vetor diário de preços futuros. O segundo núcleo recebe as previsões de preços das redes neurais, transforma-as em previsões de retorno e as utiliza para o cálculo do vetor diário de previsões Q . Então,

utilizando a equação (5.5) junto à matriz Ω – obtida por dados históricos de performance das redes especialistas – e o retorno de equilíbrio diário Π_i , calcula os retornos *a posteriori*. Os retornos *a posteriori*, ou retornos de Black-Litterman, são utilizados na equação de otimização de Markowitz sem restrições para calcular a carteira diária w'_i . A figura 19 ilustra o funcionamento do sistema em diagrama de blocos. O sistema de Benchmark, por sua vez, utiliza também uma rede neural especializada para operar o índice Bovespa, mas não possui um segundo núcleo nem utiliza a otimização de carteiras.

Figura 19: Diagrama do Núcleo 2

6 RESULTADOS

As carteiras obtidas pela otimização sem restrições, em sua maioria, utilizam a estratégia *Long and Short*, se expondo negativamente a alguns ativos (posição vendida) e positivamente a outros (posição comprada). Desta maneira não seria incomum se o modelo acabasse negociando um valor financeiro acima do seu patrimônio, ficando alavancado. Porém, para a aversão ao risco utilizada ($\delta = 2.5$), e para a metodologia utilizada para o cálculo da matriz Ω , as carteiras produzidas se mostraram bastante conservadoras. Considerando a prática comum entre corretoras brasileiras de tomar uma garantia financeira para posições vendidas, e assumindo uma garantia financeira de valor igual à quantia investida *intraday*, pode-se calcular a exposição ao mercado como sendo a somatória dos valores absolutos das posições compradas e vendidas de uma carteira, ou seja: $\sum_i |w'_i|$.

Assim, a exposição máxima da carteira ao mercado foi de 72% em 05/12/2014. A exposição mínima foi de 19% em 14/07/2015, e a exposição média foi de 35%. A figura 20 mostra a frequência em dias dos índices de exposição ao mercado produzidos pela otimização sem restrições.

Figura 20: Exposição ao mercado

Tanto no modelo proposto, quanto no modelo de *Benchmark*, o dinheiro não aplicado na carteira sugerida foi alocado no ativo livre de risco. A taxa de rendimento de depósito

interbancário (CDI), índice de boa rentabilidade no período onde o modelo foi testado, foi escolhida como indexador do ativo livre de risco. Foi utilizada a taxa do CDI exercida ao longo do período de verificação – 30/04/2014 à 29/04/2016. Em relação às posições vendidas, onde o investidor adota uma posição 'negativa', adotou-se prática comumente utilizada em corretoras brasileiras. Para uma posição vendida onde o investidor possui valor em caixa, a quantia investida é tomada como garantia pela corretora até a liquidação da operação, e caso o investidor não possua o valor em caixa a quantia investida é tomada como empréstimo sujeito a juros diários, com a posição sendo considerada alavancada. Como em nenhum momento os sistemas adotaram posições de alavancagem, adotou-se o primeiro caso.

Nestas condições, para uma carteira w , a valorização diária de patrimônio produzida pelo modelo é:

$$\begin{aligned} V_i &= w_i \cdot \Delta P_i + \bar{w}_i \Delta P_{rf} \\ w &= w_i + \bar{w}_i \end{aligned} \quad (6.1)$$

Onde ΔP_{rf} é a valorização proveniente do ativo livre de risco, ΔP_i o vetor de variações de preços dos ativos que compõem a carteira, w_i a carteira produzida pela otimização sem restrições e $\bar{w}_i$ o excedente não aplicado na carteira.

A figura 21 apresenta a evolução de patrimônio produzida pelo modelo em comparação à evolução produzida pelo sistema de *Benchmark*, com o índice Bovespa no mesmo período para referência.

A comparação dos desempenhos sob a ótica de risco e retorno pode ser feita pelas métricas da Tabela 9:

	Retorno Anualizado	Volatilidade	Índice de Sharpe	DrawDown Máximo
Modelo	47.05%	7.73%	4.45	-5.13%
Benchmark	43.96 %	14.24%	2.20	-13.53%
IBOV	1.98%	26.35%	-0.41	-47.07%

Tabela 9: Métricas de Desempenho

Figura 21: Curva de Evolução de Patrimônio

Destaca-se o resultado mais positivo ao final do período analisado, de janeiro à abril de 2016 – figura 22. A figura 23 apresenta a média de pesos produzidos pelo otimizador e a média de retornos de cada ativo, por mês analisado, onde é possível observar este comportamento com mais detalhes. Comparando a valorização com índice DI no período, o modelo produziu um retorno de aproximadamente 175% do DI.

Figura 22: Curva de Evolução de Patrimônio de Janeiro a Abril de 2016

Figura 23: Médias de pesos das carteiras no mês (vermelho) e dos retornos diários (azul)

7 CONCLUSÃO

Tanto o sistema proposto quanto o *Benchmark* apresentaram ótima performance em relação ao IBovespa. Porém, o sistema proposto apresentou um comportamento bastante superior ao *Benchmark*, apresentando um índice de Sharpe com o dobro do valor e um DrawDown Máximo significativamente menor. Validando assim o argumento a favor da abordagem utilizando inferência Bayesiana, e se mostrando uma ferramenta interessante no auxílio à gestão de carteiras. A principal vantagem do *Benchmark* em relação ao modelo se deve à menor quantidade de operações necessárias, o que diminuiria os custos operacionais em uma aplicação prática.

Bons resultados podem ser obtidos em outras escalas de previsões. O trabalho de Gambogi e Costa (2013) utiliza posições semanais utilizando previsões de redes neurais, o que reduziria os custos operacionais e aumentaria a viabilidade de implantação prática. Outra limitação do modelo é a metodologia de busca exaustiva em si, que demandou um enorme esforço computacional, mesmo com as simplificações adotadas. Outras abordagens para otimização de treinamento, como a utilização de algoritmos genéticos aplicados na seleção de entradas (*features*) ou de redes vencedoras, poderiam produzir os mesmos resultados com menor tempo de execução.

O resultado mais interessante, entretanto, envolve as taxas de acerto das redes neurais. Nenhuma taxa de acerto individual das redes se manteve acima de 60%, resultado não muito impressionante mas, quando utilizadas em conjunto com a inferência bayesiana do modelo de Black-Litterman, apresentaram um ótimo retorno anualizado com baixa volatilidade, refletidos no índice de Sharpe. Foi observado que as taxas de acerto de muitas redes aumentaram em determinadas janelas no período de teste. Poderia-se aproveitar este comportamento atualizando-se a confiabilidade da equação (3.6) de maneira adaptativa, produzindo assim uma matriz Ω variável.

A etapa de treinamento das redes exigiu enorme esforço computacional, mesmo para uma quantidade reduzida de ativos e um intervalo de treinamento e testes relativamente curto. Trabalhos futuros podem endereçar este problema de maneira mais eficiente, uti-

lizando, por exemplo, abordagens de computação evolutiva para seleção de arquiteturas e redes vencedoras ou outros algoritmos de otimização – as redes deste trabalho utilizaram o otimizador ADAM – o que possibilitaria o uso de redes com mais neurônios nas camadas escondidas, e técnicas de treinamento *on-line*. Outra sugestão para avaliar o comportamento das redes neurais é utilizar outras abordagens para compor o sistema de benchmark, como sistemas baseados em ARMA-GARCH para previsão de variações de preços.

Como alternativas para as entradas das redes, podem ser feitos estudos com outras abordagens além de indicadores de análise técnica, como indicadores macroeconômicos, ou análises de book de ofertas (*tape-reading*). Entre outras sugestões para substituição do primeiro núcleo incluem-se redes de *deep-learning* para reconhecimento de imagens na busca de padrões gráficos de *candles* ou fractais e técnicas de análise comportamental através de sistemas de *machine-learning* que monitorem notícias e redes sociais. Como última sugestão, poderia-se implementar o método CAPM, ou outras abordagens de estimativa de preços, para encontrar os retornos *a priori* – ou retornos de equilíbrio – dando mais liberdade para implementação do sistema para ativos que não compõe o índice Bovespa.

REFERÊNCIAS

- BLACK, F.; LITTERMAN, R. Global portfolio optimization. *Financial analysts journal*, CFA Institute, v. 48, n. 5, p. 28–43, 1992.
- BLACK, F.; LITTERMAN, R. B. Asset allocation: combining investor views with market equilibrium. *The Journal of Fixed Income*, Institutional Investor Journals, v. 1, n. 2, p. 7–18, 1991.
- CHOLLET, F. et al. *Keras*. 2015. <<https://keras.io>>.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989.
- ELMAN, J. L. Finding structure in time. *Cognitive science*, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990.
- FABOZZI, F. J.; FOCARDI, S. M.; KOLM, P. N. Incorporating trading strategies in the black-litterman framework. *The Journal of Trading*, Institutional Investor Journals, v. 1, n. 2, p. 28–37, 2006.
- GAMBOGI, J. A.; COSTA, O. L. do V. Sistema de trading no mercado de ações implementado por redes neurais artificiais. *Learning & Nonlinear Models*, SBRN, v. 11, n. 2, p. 92–102, 2013.
- HE, G.; LITTERMAN, R. The intuition behind black-litterman model portfolios. *Investment Management Research*, Goldman, Sachs & Company, 10 2002.
- IDZOREK, T. A step-by-step guide to the black-litterman model: Incorporating user-specified confidence levels. In: *Forecasting expected returns in the financial markets*. [S.l.]: Elsevier, 2007. p. 17–38.
- JEGADEESH, N.; TITMAN, S. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of finance*, Wiley Online Library, v. 48, n. 1, p. 65–91, 1993.
- JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001. <<http://www.scipy.org/>>. [Online; accessed 26 november 2018].
- JORDAN, M. I. Serial order: A parallel distributed processing approach. In: *Advances in psychology*. [S.l.]: Elsevier, 1997. v. 121, p. 471–495.
- KAASTRA, I.; BOYD, M. Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, Elsevier, v. 10, n. 3, p. 215–236, 1996.
- KIMOTO, T. et al. Stock market prediction system with modular neural networks. In: IEEE. *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. [S.l.], 1990. p. 1–6.

- KLASSEN, M. Investigation of some technical indexes instock forecasting using neural networks. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, World Academy of Science, Engineering and Technology, v. 1, n. 5, p. 1438–1442, 2007.
- KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, IEEE, v. 78, n. 9, p. 1464–1480, 1990.
- LEE, W. *Theory and methodology of tactical asset allocation*. [S.l.]: John Wiley & Sons, 2000. v. 65.
- LO, A. W. The adaptive markets hypothesis. *The Journal of Portfolio Management*, Institutional Investor Journals, v. 30, n. 5, p. 15–29, 2004.
- MACIEL, L. S.; BALLINI, R. Neural networks applied to stock market forecasting: An empirical analysis. *Journal of the Brazilian Neural Network Society*, v. 8, n. 1, p. 3–22, 2010.
- MARKOWITZ, H. Portfolio selection. *The journal of finance*, Wiley Online Library, v. 7, n. 1, p. 77–91, 1952.
- MARTINEZ, L. C. et al. From an artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa. In: IEEE. *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. [S.l.], 2009. p. 2006–2013.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- MEUCCI, A. Beyond black-litterman in practice: A five-step recipe to input views on non-normal markets. *SSRN Electronic Journal*, v. 19, 05 2006.
- MICHAUD, R. O. The markowitz optimization enigma: Is ‘optimized’ optimal? *Financial Analysts Journal*, CFA Institute, v. 45, n. 1, p. 31–42, 1989.
- MINSKY, M.; PAPERT, S. *Perceptrons*. MIT Press, 1969.
- MURPHY, J. J. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. [S.l.]: Penguin, 1999.
- OLIPHANT, T. E. *A guide to NumPy*. [S.l.]: Trelgol Publishing USA, 2006. v. 1.
- PETERS, E. E.; PETERS, E. R.; PETERS, D. *Fractal market analysis: applying chaos theory to investment and economics*. [S.l.]: John Wiley & Sons, 1994. v. 24.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986.
- RUMELHART, D. E. et al. *Parallel distributed processing*. [S.l.]: MIT press Cambridge, MA, 1987. v. 1.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Malaysia; Pearson Education Limited,, 2016.

SATCHELL, S.; SCOWCROFT, A. A demystification of the black-litterman model: Managing quantitative and traditional portfolio construction. *Journal of Asset Management*, Palgrave Macmillan, v. 1, n. 2, p. 138–150, 2000.

SHARPE, W. F. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, Wiley Online Library, v. 19, n. 3, p. 425–442, 1964.

SILVER, D. et al. Mastering the game of go without human knowledge. *Nature*, Nature Publishing Group, v. 550, n. 7676, p. 354, 2017.

THEIL, H. *Principles of econometrics*. [S.l.], 1971.

TURING, A. M. Computing machinery and intelligence. *Mind*, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Disponível em: <https://dx.doi.org/10.1093/mind/LIX.236.433>.

WALTERS, J. The black-litterman model in detail (june 20, 2014). Available at SSRN: <https://ssrn.com/abstract=1314585> or <http://dx.doi.org/10.2139/ssrn.1314585>. 2014.

WHITE, H. Economic prediction using neural networks: The case of ibm daily stock return. In: *Proceedings of International Conference on Neural Networks*. [S.l.: s.n.], 1988. p. II–451.

WUTSQA, D. U.; KUSUMAWATI, R. S. R. Radial basis function neural network for views prediction on black-litterman model. *Journal of Innovative Technology and Education*, v. 3, n. 1, p. 71–78, 2016.

XING, F. Z.; CAMBRIA, E.; WELSCH, R. E. Intelligent asset allocation via market sentiment views. *IEEE Computational Intelligence Magazine*, IEEE, v. 13, n. 4, p. 25–34, 2018.

ZIMMERMANN, H. G.; NEUNEIER, R.; GROTHMANN, R. Active portfolio-management based on error correction neural networks. In: MIT PRESS. *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. [S.l.], 2001. p. 1465–1472.

ANEXO A – DEMONSTRAÇÕES MATEMÁTICAS

A.1 Retorno esperado de Black-Litterman

Este apêndice apresenta a dedução matemática da equação (3.4), a partir da equação (3.1), como visto em Walters (2014):

$$\begin{aligned}
\hat{\Pi} &= [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P'\Omega^{-1}Q] \\
\hat{\Pi} &= [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}(\tau\Sigma)^{-1}\Pi + [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}(P'\Omega^{-1}Q) \\
\hat{\Pi} &= [[\tau\Sigma - \tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\tau\Sigma](\tau\Sigma)^{-1}\Pi] + [[(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}(P'\Omega^{-1}Q)] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [[(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}(P'\Omega^{-1}Q)] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [(\tau\Sigma)(\tau\Sigma)^{-1}[(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}(P'\Omega^{-1}Q)] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [(\tau\Sigma)[I_n + P'\Omega^{-1}P\tau\Sigma]^{-1}(P'\Omega^{-1}Q)] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [\tau\Sigma[I_n + P'\Omega^{-1}P\tau\Sigma]^{-1}(P'\Omega^{-1}Q)] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [\tau\Sigma[I_n + P'\Omega^{-1}P\tau\Sigma]^{-1}(\Omega(P')^{-1})^{-1}Q] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [\tau\Sigma[\Omega(P')^{-1} + P\tau\Sigma]^{-1}Q] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [\tau\Sigma P'(P')^{-1}[\Omega(P')^{-1} + P\tau\Sigma]^{-1}Q] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [\tau\Sigma P'[\Omega + P\tau\Sigma P']^{-1}Q] \\
\hat{\Pi} &= [\Pi - [\tau\Sigma P'[P\tau\Sigma P' + \Omega]^{-1}P\Pi]] + [\tau\Sigma P'[\Omega + P\tau\Sigma P']^{-1}Q] \\
\hat{\Pi} &= \Pi + \tau\Sigma P'[(P\tau\Sigma P') + \Omega]^{-1}[Q - P\Pi]
\end{aligned}$$

c.q.d.

A.2 Método de Theil

Este apêndice apresenta o desenvolvimento matemático da equação de Black Litterman utilizando a abordagem de Theil –Theil (1971)– para o problema de média desconhecida e variância conhecida com informações parciais:

A média da distribuição dos retornos *a priori* é π com valor esperado β e um valor residual u :

$$\pi = x\beta + u \quad (\text{A.1})$$

Onde u é uma distribuição normal de média 0 e variância Φ .

A informação adicional, uma distribuição de probabilidade condicional, é dada por:

$$q = p\beta + v \quad (\text{A.2})$$

com v possuindo média 0 e variância Ω . Assume-se que tanto Ω quanto Φ são não singulares.

A combinação dos valores *a priori* e informações condicionais é então:

$$\begin{bmatrix} \pi \\ q \end{bmatrix} = \begin{bmatrix} x \\ p \end{bmatrix} \beta + \begin{bmatrix} u \\ v \end{bmatrix}$$

O valor esperado do residual é igual à zero, e a variância é:

$$E \left(\begin{bmatrix} u \\ v \end{bmatrix} \begin{bmatrix} u' & v' \end{bmatrix} \right) = \begin{bmatrix} \Phi & 0 \\ 0 & \Omega \end{bmatrix}$$

Então, pelo método dos mínimos quadrados generalizados, a estimativa de β pode ser descrita por:

$$\hat{\beta} = \begin{bmatrix} x & p \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Omega \end{bmatrix} \begin{bmatrix} x' \\ p' \end{bmatrix}^{-1} \begin{bmatrix} x' & p' \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Omega \end{bmatrix}^{-1} \begin{bmatrix} \pi \\ q \end{bmatrix}$$

Que, reescrita sem a notação matricial resulta em:

$$\hat{\beta} = [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1} [x'\Phi^{-1}\pi + p'\Omega^{-1}q] \quad (\text{A.3})$$

Equivalente à equação (3.1).

Pode-se chegar à equação (3.2) de forma análoga, considerando que a variância pode ser calculada pelo valor esperado de $(\hat{\beta} - \beta)^2$, como segue.

Substituindo as equações A.1 e A.2 e rearranjando:

$$\begin{aligned}\hat{\beta} &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}[x'\Phi^{-1}(x\beta + u) + p'\Omega^{-1}(p\beta + v)] \\ \hat{\beta} &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}[x'\Phi^{-1}x\beta + x'\Phi^{-1}u + p'\Omega^{-1}p\beta + p'\Omega^{-1}v] \\ \hat{\beta} &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}[x\Phi^{-1}x'\beta + p\Omega^{-1}p'\beta] + [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}[x\Phi^{-1}u + p\Omega^{-1}v] \\ \hat{\beta} &= \beta + [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}[x\Phi^{-1}u + p\Omega^{-1}v]\end{aligned}$$

Então, calculando o valor esperado:

$$\begin{aligned}\hat{\beta} - \beta &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}[x\Phi^{-1}u + p\Omega^{-1}v] \\ E[(\hat{\beta} - \beta)^2] &= E[[x\Phi^{-1}x' + p\Omega^{-1}p']^{-2}[x\Phi^{-1}u'u\Phi^{-1}x' + \\ &\quad + p\Omega^{-1}v'v\Omega^{-1}p' + x\Phi^{-1}u'v\Omega^{-1}p' + p\Omega^{-1}v'u\Phi^{-1}x']]\end{aligned}$$

Porém, temos que $E(uu') = \Phi$, $E(vv') = \Omega$, $E(uv') = 0$, então:

$$\begin{aligned}E[(\hat{\beta} - \beta)^2] &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-2}[(x\Phi^{-1}\Phi\Phi^{-1}x') + (p\Omega^{-1}\Omega\Omega^{-1}p') + 0 + 0] \\ E[(\hat{\beta} - \beta)^2] &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-2}[(x\Phi^{-1}x') + (p\Omega^{-1}p')] \\ E[(\hat{\beta} - \beta)^2] &= [x\Phi^{-1}x' + p\Omega^{-1}p']^{-1}\end{aligned}$$

Para o modelo de Black Litterman, Φ é proporcional a Σ e x é a matriz identidade, então:

$$\begin{aligned}\Phi &= \tau\Sigma, \quad x = I \\ M &= [(\tau\Sigma)^{-1} + p\Omega^{-1}p']^{-1}\end{aligned}$$