



SELEÇÃO DE PORTFÓLIOS USANDO REDES NEURAIAS ADVERSÁRIAS
GENERATIVAS PARA A MODELAGEM DO PROCESSO ESTOCÁSTICO DE
RETORNOS

Rafael Lopes Conde dos Reis

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Felipe Maia Galvão França
Flávio Luis de Mello

Rio de Janeiro
Janeiro de 2019

SELEÇÃO DE PORTFÓLIOS USANDO REDES NEURAS ADVERSÁRIAS
GENERATIVAS PARA A MODELAGEM DO PROCESSO ESTOCÁSTICO DE
RETORNOS

Rafael Lopes Conde dos Reis

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE
SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Felipe Maia Galvão França, Ph.D.

Prof. Flávio Luis de Mello, D.Sc.

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Prof. Priscila Machado Vieira Lima, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2019

Lopes Conde dos Reis, Rafael

Seleção de Portfólios Usando Redes Neurais Adversárias Generativas para a Modelagem do Processo Estocástico de Retornos/Rafael Lopes Conde dos Reis. – Rio de Janeiro: UFRJ/COPPE, 2019.

XVIII, 94 p.: il.; 29, 7cm.

Orientadores: Felipe Maia Galvão França

Flávio Luis de Mello

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 79 – 86.

1. Seleção de Portfólios. 2. Redes Neurais. 3. Redes Neurais Adversárias Generativas. 4. Mercado de Ações. I. Maia Galvão França, Felipe *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*"Ideias e somente ideias podem
iluminar a escuridão"*
Ludwig von Mises

Agradecimentos

Gostaria de agradecer aos meus pais que me incentivaram a ingressar no mestrado. Agradeço a minha família que sempre me estimulou a continuar a me dedicar neste projeto, mesmo com todas as dificuldades de conciliar minha pesquisa e meu trabalho. Agradeço a minha namorada por sempre me apoiar, mesmo as custas de muitos finais de semana, e por me vigiar para garantir que eu estava focado. Agradeço aos meus sócios por respeitarem esta escolha e por terem paciência, quando em muitos momentos nem eu mesmo tinha. Agradeço aos meus orientadores, por acreditarem quando eu sugeri mudar o rumo da minha pesquisa e por continuarem me orientando mesmo depois de tanto tempo. Agradeço também a todos os meus amigos por compreenderem todas as vezes que precisei estar ausente. Por ultimo agradeço a minha cabeça dura de não desistir de um desafio, mesmo quando, em muitos momentos, esta parecesse ser uma boa decisão.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SELEÇÃO DE PORTFÓLIOS USANDO REDES NEURAS ADVERSÁRIAS GENERATIVAS PARA A MODELAGEM DO PROCESSO ESTOCÁSTICO DE RETORNOS

Rafael Lopes Conde dos Reis

Janeiro/2019

Orientadores: Felipe Maia Galvão França
Flávio Luis de Mello

Programa: Engenharia de Sistemas e Computação

A diversificação de ativos financeiros é uma forma de reduzir a exposição ao risco mantendo uma taxa de retorno satisfatória. Os métodos de seleção de portfólios de ativos, tradicionalmente usados, exigem fortes simplificações para modelar a dinâmica de seus preços. Tais simplificações incluem desconsiderar relações não-lineares na composição dos preços dos ativos e até a suposição de que o processo estocástico que os compõem seja estacionário. Redes neurais têm a capacidade de modelar funções não lineares e são amplamente usadas para a estimação de preços de ações, mas por serem estimativas determinísticas estes modelos acabam sendo de pouco uso no problema de seleção e portfólios.

Redes Generativas Adversárias são o estado da arte para a modelagem de distribuições de probabilidade complexas. Através de uma competição entre uma rede geradora e uma discriminadora é possível gerar amostras estatisticamente semelhantes a distribuição que se deseja modelar. Este artigo apresenta a aplicação de Redes Recorrentes Adversárias para modelar a distribuição conjunta de um portfólio de ações e gerar amostras para o problema de otimização de portfólio. O modelo proposto tem um retorno médio diário 34,55% superior ao melhor dos benchmarks avaliados, em dados de um conjunto de teste.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PORTFOLIO SELECTION USING RECURRENT GENERATIVE
ADVERSARIAL NETWORKS FOR MODELING RETURNS' STOCHASTIC
PROCESS

Rafael Lopes Conde dos Reis

January/2019

Advisors: Felipe Maia Galvão França
Flávio Luis de Mello

Department: Systems Engineering and Computer Science

Diversification of financial assets is a way to reduce risk exposure while maintaining a satisfactory rate of return. Traditionally used asset portfolio selection methods require strong simplifications in order to model price dynamics. Such simplifications include disregarding nonlinear relationships in the composition of asset prices and even the assumption that the stochastic process of asset's returns is stationary. Neural networks have the ability to model nonlinear functions and are widely used for stock price estimation, but because these estimations are deterministic, these models end up being of little usage for portfolio selection.

Generative Adversarial Networks are the state of the art for modeling complex probability distributions. Through a competition between a generator and a discriminator network it is possible to generate samples statistically similar to the desired distribution. This research presents the application of Generative Recurrent Adversarial Networks to model the joint distribution of a portfolio and generate samples for the optimization problem. The proposed model has an average daily return 34.55% higher than the best among the evaluated benchmarks on out of sample data.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiii
Lista de Símbolos	xv
Lista de Abreviaturas	xvii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos e Contribuições	2
1.3 Estrutura da Dissertação	3
2 Seleção de Portfólio	4
2.1 Modelo Média-Variância	4
2.1.1 História da Teoria Moderna do Portfólio	4
2.1.2 Problema de Otimização dos Pesos	5
2.1.3 Fronteira Eficiente	8
2.1.4 Ativo Livre de Risco e o Portfólio Tangente	10
2.1.5 Deficiências do Modelo Média-Variância	13
2.2 Propostas para a Estimativa dos Parâmetros	14
2.2.1 Modelo de Índice Único	15
2.2.2 Modelos de Multi-Fator	15
2.2.3 Métodos de Encolhimento	16
3 Redes Adversárias Generativas Aplicadas a Séries Temporais	19
3.1 Redes Neurais Artificiais	19
3.1.1 Neurônio Artificial	19
3.1.2 Backpropagation e o Perceptron Multicamadas	22
3.2 Redes Neurais Recorrentes	27
3.2.1 Elman e Jordan	28
3.2.2 Treinamento de RNNs e o Problema de Longas Dependências	29

3.2.3	Longa Memória de Curto Prazo	30
3.2.4	RNNs para Aplicações em Séries Temporais Financeiras	32
3.3	Redes Generativas Adversárias	33
3.3.1	Wasserstein GAN	36
3.3.2	Discriminação de Mini-lotes	38
3.3.3	Outras Técnicas para o Treinamento de GANs	39
3.3.4	GANs Aplicadas a Séries Temporais	40
4	Arquitetura do Modelo e Viabilidade	42
4.1	RNNs para a Geração de Séries Temporais	43
4.2	GANs para a Geração de Séries Temporais	46
4.2.1	Dados Sintéticos	46
4.2.2	GAN Composta por LSTM	47
4.2.3	GAN de Recorrência Codificada	52
4.2.4	Comparação entre Métodos de Seleção de Portfólio	55
5	Aplicação a Dados de Ativos Reais	57
5.1	Dados e Variáveis Utilizados	57
5.1.1	Um Ativo por Indústria	57
5.1.2	100 Ativos do S&P 500	58
5.1.3	Análise das Variáveis	59
5.2	Metodologia do Experimento	60
5.2.1	Ajuste de Hiperparâmetros	62
5.2.2	Avaliação de Técnicas para o Treinamento da GAN	64
5.2.3	Comparação entre Métodos de Seleção de Portfólio	64
5.3	Resultados e Discussões	66
5.3.1	Um Ativo por Indústria	66
5.3.2	100 Ativos do S&P 500	71
6	Conclusões	78
	Referências Bibliográficas	79
A	Evolução do Treinamento com Dados Sintéticos	87
B	Ativos Presentes no Dataset Um Ativo por Indústria	90
C	Ativos Presentes no Dataset 100 Ativos do S&P 500	91

Lista de Figuras

2.1	Relação entre o valor esperado de retornos e risco	9
2.2	Fronteira eficiente em um universo de quatro ativos	10
2.3	Fronteiras eficientes para diferentes valores de correlação	10
2.4	Fronteira eficiente com um ativo livre de risco	11
2.5	Fronteira eficiente em um universo de quatro ativos e um ativo livre de risco	13
3.1	Modelo McCulloch-Pitts de um neurônio artificial	20
3.2	Modelo Perceptron de um neurônio artificial	20
3.3	Modelo Perceptron Multicamada com 3 entradas, 4 neurônios na camada escondida e 2 neurônios na camada de saída	22
3.4	Rede Neural Recorrente	27
3.5	Rede Elman	28
3.6	Rede Jordan	29
3.7	RNN desdobrada com $\tau = 3$	29
3.8	Célula LSTM	30
3.9	Rede Generativa Adversária	34
4.1	Distribuição do Processo Estocástico Gaussiano Estacionário no instante t	44
4.2	Ocorrência do Processo Estocástico Gaussiano Estacionário	44
4.3	Evolução do treinamento da LSTM	45
4.4	Série temporal gerada por uma LSTM	45
4.5	Distribuição das amostras geradas por uma LSTM	46
4.6	Séries temporais dos ativos de um portfólio sintético	47
4.7	Distribuição de probabilidade conjunta dos ativos de um portfólio num instante de tempo t	48
4.8	Modelo GAN para séries temporais	49
4.9	Distribuição dos dados gerados ao longo do tempo	50
4.10	Distribuição dos dados gerados no instante $(t + 1)$	51
4.11	Modelo GAN condicionado por uma camada LSTM	52

4.12	Distribuição dos dados gerados pela ER-GAN no instante $(t + 1)$. . .	53
4.13	Comparação entre diferentes tamanhos de janela	54
4.14	Comparação entre diferentes taxas de aprendizado	54
4.15	Comparação entre a ER-GAN, o modelo de Média-Variância e um portfólio igualmente balanceado	56
5.1	Distribuição conjunta dos retornos dos ativos presentes no portfólio .	61
5.2	Média e variância dos retornos em janelas de 200 amostras	62
5.3	Distribuição de um ativo do portfólio em diferentes janelas de tempo	62
5.4	Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN	67
5.5	Distribuição conjunta real e gerada pela ER-GAN de dois ativos do portfólio	68
5.6	Comparação entre taxas de aprendizado para o dataset Um Ativo por Indústria	68
5.7	Comparação entre tamanhos de janela para o dataset Um Ativo por Indústria	69
5.8	Comparação entre métodos de treinamento de GAN para o dataset Um Ativo por Indústria	69
5.9	Comparação do retorno acumulado entre métodos para a melhoria da diversidade das amostras	70
5.10	Comparação do retorno acumulado entre a melhor configuração de ER-GAN e os benchmarks	71
5.11	Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN com o dataset 100 Ativos do S&P 500	73
5.12	Distribuição conjunta real e gerada pela ER-GAN de dois ativos do portfólio para o dataset 100 Ativos do S&P 500	73
5.13	Comparação entre taxas de aprendizado para o dataset 100 Ativos do S&P 500	74
5.14	Comparação entre tamanhos de janela para o dataset 100 Ativos do S&P 500	74
5.15	Comparação entre métodos de treinamento de GAN para o dataset 100 Ativos do S&P 500	75
5.16	Comparação do retorno acumulado entre métodos para a melhoria da diversidade das amostras com o dataset 100 Ativos do S&P 500 . . .	76
5.17	Comparação do retorno acumulado entre a melhor configuração de ER-GAN e os benchmarks para o dataset 100 Ativos do S&P 500 . .	77
A.1	Evolução da qualidade das amostras geradas pela ER-GAN para dados sintéticos	87

A.2	Evolução da qualidade das amostras geradas pela ER-GAN para o dataset Um Ativo por Indústria	88
A.3	Evolução da qualidade das amostras geradas pela ER-GAN para o dataset 100 Ativos do S&P 500	89

Lista de Tabelas

3.1	Exemplos de funções de ativação comumente utilizadas. u representa a entrada da função de ativação, ou seja, $u = w \odot x + b$.	21
3.2	Equações do Algoritmo de Backpropagation	26
5.1	Estatística dos dados utilizados	60
5.2	Configurações de topologia avaliadas por busca aleatória	63
5.3	Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN	66
5.4	Comparação entre taxas de aprendizado para o dataset Um Ativo por Indústria	67
5.5	Comparação entre tamanhos de janela para o dataset Um Ativo por Indústria	69
5.6	Comparação entre métodos de treinamento de GAN para o dataset Um Ativo por Indústria	69
5.7	Comparação entre métodos para a melhoria da diversidade das amostras com o dataset Um Ativo por Indústria	70
5.8	Comparação entre o modelo ER-GAN e benchmarks para o dataset Um Ativo por Indústria	70
5.9	Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN com o dataset de 100 Ativos do S&P 500	72
5.10	Comparação entre taxas de aprendizado para o dataset 100 Ativos do S&P 500	74
5.11	Comparação entre tamanhos de janela para o dataset 100 Ativos do S&P 500	74
5.12	Comparação entre métodos de treinamento de GAN para o dataset 100 Ativos do S&P 500	75
5.13	Comparação entre métodos para a melhoria da diversidade das amostras com o dataset 100 Ativos do S&P 500	75
5.14	Comparação entre o modelo ER-GAN e benchmarks para o dataset 100 Ativos do S&P 500	76

B.1	Ações do dataset Um Ativo por Indústria	90
C.1	Ações do dataset 100 Ativos do S&P 500	91

Lista de Símbolos

C	Função custo, p. 22
F	Total de fatores em um modelo multi-fator, p. 16
M	Portfólio Tangente, p. 11
N	Total de ativos em um portfólio, p. 6
$R_i(t)$	Variável aleatória do retorno de um ativo i no instante de tempo t , p. 6
$R_p(t)$	Variável aleatória do retorno do portfólio t , p. 6
T	Total de amostras na série temporal, p. 6
α_i	Valor do retorno anormal de um ativo, p. 15
β_i	Taxa de sensibilidade de um ativo ao mercado, p. 15
$\beta_{i,f}$	Taxa de sensibilidade de um ativo a um determinado fator, p. 16
Σ	Matriz de covariância dos ativos de um portfólio, p. 7
μ_t	Vetor do valor esperado do retorno dos ativos de um portfólio, p. 7
w_M^*	Vetor de pesos do portfólio tangente, p. 13
w	Vetor dos pesos dos ativos i de um portfólio, p. 7
δ_j^l	Erro no neurônio j da camada l , p. 23
$\hat{\Sigma}$	Matriz de covariância amostral dos ativos em um portfólio, p. 14
$\hat{\mu}$	Média amostral dos retornos de um ativo, p. 14
\mathcal{F}_f	Valor de um determinado fator num instante de tempo, p. 16

μ_0	Retorno do ativo livre de risco, p. 11
$\mu_M(t)$	Valor esperado de retorno do portfólio tangente no instante t , p. 13
$\sigma_M^2(t)$	Variância do portfólio tangente no instante t , p. 13
σ_j^l	Erro do neurônio j da camada l , p. 23
a_j^l	Ativação no neurônio j da camada l , p. 23
$cov_{i,j}(t)$	Covariância entre dois ativos i e j no instante t , p. 6
h^t	Estado da rede neural no instante t , p. 27
i	Índice do ativos em um portfólio, p. 6
$p_i(t)$	Preço de um ativo no instante de tempo t , p. 6
$r_i(t)$	Retorno de um ativo no instante de tempo t , p. 6
$r_m(t)$	Retorno do mercado em um instante de tempo t , p. 15
t	Índice do instante de tempo, p. 6
w_i	Peso de um ativo i em um portfólio, p. 6
z_j^l	Entrada da função de ativação no neurônio j da camada l , p. 23

Lista de Abreviaturas

B Norm	Batch Normalization, p. 64
CCS	Constant Correlation Shrinkage, p. 65
CNN	Convolutional Neural Network, p. 34
CVS	Constant Variance Shrinkage, p. 65
DCGAN	Deep Convolutional Generative Adversarial Network, p. 38
Divergência KL	Divergência de Kullback-Leibler, p. 36
ER-GAN	Encoded Recurrence GAN, p. 52
EW	Equally Weighted, p. 65
F/A	Razão fechamento/abertura, p. 59
F/Max	Razão fechamento/máxima, p. 59
F/Min	Razão fechamento/mínima, p. 59
GAN	Generative Adversarial Network, p. 2
GARCH	Generalized Autoregressive Conditional Heteroskedasticity, p. 33
GRU	Gated Recurrent Unit, p. 32
HME	Hipótese do Mercado Eficiente, p. 46
Leaky ReLU	Leaky Rectified Linear Unit, p. 21
MB Disc	Minibatch Discrimination, p. 64
MLP	Multi Layer Perceptron, p. 22
MSE	Mean Squared Error, p. 17
MV	Mean-Variance, p. 65

OAS	Oracle Approximating Shrinkage, p. 65
RNN	Recurrent Neural Network, p. 26
ReLU	Rectified Linear Unit, p. 21
SFS	Single Factor Shrinkage, p. 65
SVM	Suport Vector Machines, p. 33
VV	Variação do volume, p. 59
WGAN-GP	Wasserstein GAN with Gradient Penalty, p. 38
WGAN	Wasserstein Generative Adversarial Network, p. 36

Capítulo 1

Introdução

1.1 Motivação

Indivíduos buscam aumentar a quantidade de recursos que terão disponíveis no futuro por meio de uma alocação inteligente destes no presente. Chamamos estes recursos de ativos e chamamos esta alocação de investimento. Toda modalidade de investimento está sujeita a um fator de risco, uma vez que não é possível definir com certeza o futuro. Cabe a um investidor racional sempre buscar um investimento que possua um maior valor esperado de retorno associado a um menor risco.

Um portfólio consiste na combinação de um universo de ativos como forma de investimento. Ao variarmos a proporção investida em cada ativo, é possível obter um valor diferente de retorno associado a um diferente nível de risco. Isso se deve ao fato de que cada ativo tem, individualmente, seu próprio valor esperado de retorno e nível de risco. O valor investido em cada ativo é definido de forma a melhor atender o grau de tolerância a risco de um investidor. Este processo de definir a melhor alocação do total investido, entre os diferentes ativos é chamado de Seleção de Portfólio.

O modelo de média-variância proposto por Markowitz [1, 2] foi um marco na Teoria Moderna de Portfólio, tendo lhe concedido um Prêmio Nobel de Economia em 1990. Consiste em uma maneira de construir um portfólio de forma a maximizar seu retorno para um dado nível de risco. Ele considera um portfólio eficiente todo aquele portfólio que possui o maior retorno possível, para um determinado nível de risco, dado um universo de ativos que o compõe. Markowitz propõe o uso da variância como medida de risco e estima o valor esperado e a matriz de covariância dos retornos por meio do uso da média e covariância amostrais.

Usar todas as amostras dos retornos para a estimativa dos parâmetros não é algo viável, uma vez que a série temporal de retornos não é estacionária, e seu valor esperado e matriz de correlação variam com o tempo. Por outro lado, limitar o

numero de amostra apenas às mais recentes faz com que a estimação seja imprecisa, especialmente para portfólios com muitos ativos. Isso levou a apresentação de diversas técnicas na literatura, buscando formas de se estimar um portfólio ótimo, como Modelos Multi-fator e Estimadores de Encolhimento. Estas técnicas, porém, fazem uso de fortes simplificações para o comportamento dos retornos e muitas vezes não apresentam resultados consistentemente superiores ao de um portfólio igualmente distribuído [3].

A maior simplificação, porém, está em utilizar as amostras temporais das séries de retornos como se elas fossem amostras de uma mesma variável aleatória. Ao fazer isso, trata-se o processo estocástico dos retornos como sendo estacionário, o que não é o caso [4, 5]. Cada amostra das séries temporais é uma ocorrência única de uma variável aleatória pertencente a um processo estocástico. Caso fosse possível ter acesso a diferentes ocorrências da variável aleatória referente a um instante da série temporal seria então possível estimar propriedades como o valor esperado e a matriz de covariâncias para aquele instante.

Redes Generativas Adversárias (Generative Adversarial Networks - GANs) [6] são o estado da arte em modelos generativos e vêm sendo utilizadas na geração de amostras vindas de diversas distribuições de probabilidade complexas (e.g. geração de imagens). GANs também possuem aplicações com séries temporais, tendo sido inclusive utilizadas para modelar o preço de ações [7]. Uma das grandes vantagens de GANs está no fato de que, uma vez treinado o modelo, a geração de amostras é extremamente barata do ponto de vista computacional. É possível então utilizar uma GAN que, uma vez recebendo o histórico de uma série temporal de retornos seja capaz de gerar amostras provenientes da distribuição do próximo instante de tempo. Desta forma pode-se utilizar estas amostras na estimação do valor esperado e da matriz de covariância deste instante.

1.2 Objetivos e Contribuições

Este trabalho tem como objetivo avaliar a viabilidade da aplicação de GANs em um problema de seleção de portfólios, a fim de gerar amostras referentes a um instante de tempo e usá-las calcular o valor esperado e matriz de covariância. Para tal, propõe-se inicialmente verificar a capacidade de modelar processos estocásticos com GANs utilizando dados provenientes de uma distribuição conhecida. Uma vez definido um modelo de GAN que consiga gerar amostras suficientemente próximas às dos processos utilizados o modelo será aplicado a dados reais de ações com o objetivo de gerar cenários para seleção de portfólios. Este modelo será avaliado com base no retorno gerado pelo portfólio construído a partir das amostras geradas por ele.

Esta é a primeira aplicação de GANs relacionada a seleção de portfólios. Propõe-se, além disso, um modelo capaz de gerar maior diversidade entre amostras para problemas de séries temporais quando comparado a outros modelos presentes na literatura. Por fim, outra contribuição do presente trabalho está em evidenciar que a qualidade dos portfólios selecionados é superior quando se utiliza um modelo que considera tanto as relação não lineares quanto o comportamento não estacionário dos processos estocásticos dos retornos dos ativos.

1.3 Estrutura da Dissertação

Esta monografia se divide em 6 capítulos. No presente capítulo apresentou-se uma breve introdução do escopo deste trabalho. No Capítulo 2 é introduzida a base teórica referente ao problema de seleção de portfólios, desde sua consolidação com os trabalhos de Markowitz, até as propostas mais recentes. No Capítulo 3 é apresentado o conceito de GANs, passando pela descrição de Redes Neurais Artificiais e sua aplicação em problemas envolvendo séries temporais financeiras. Estes dois capítulos consolidam a fundamentação teórica deste trabalho. No Capítulo 4 é descrito o modelo proposto, incluindo os experimentos realizados para avaliar a viabilidade de sua aplicação com dados sintéticos. O Capítulo 5 mostra os experimentos realizados do modelo proposto aplicado a dados reais, comparando-o a diferentes técnicas já estabelecidas para a seleção de portfólios. O Capítulo 6 conclui a análise dos resultados obtidos e indica possíveis contribuições a serem realizadas no futuro.

Capítulo 2

Seleção de Portfólio

O conceito de diversificação é bastante antigo e a ideia de que não se deve investir todo seu capital em um único ativo é um senso comum. Como diz o ditado popular, "Não se deve colocar todos os ovos na mesma cesta.". Este problema de alocação de capital em diferentes ativos é algo presente na vida de indivíduos e instituições financeiras. A este problema dá-se o nome de Teoria do Portfólio [8]. Pode-se definir um portfólio como sendo uma distribuição de capital em um determinado universo de ativos. Desta forma, um portfólio composto de 50% do ativo A e 50% do ativo B é diferente de um contendo 10% do ativo A e 90% do ativo B .

Conforme um investidor diversifica seus investimentos ele minimiza sua exposição a risco proveniente de fatores específicos a cada ativo. Por mais que se diversifique em um número muito grande de ativos, porém, é impossível evitar exposição a um fator de risco. Isto acontece, pois todos os ativos no mercado estão suscetíveis a fatores macroeconômicos comuns a eles. O risco que está sempre presente é conhecido como risco de mercado, risco sistêmico, ou risco não diversificável. Já aquele que pode ser mitigado por meio de diversificação é conhecido como risco específico, risco não sistêmico, ou risco diversificável [9].

Este capítulo irá abordar o problema de seleção de portfólios, começando com a descrição da Teoria Moderna de Portfólios. Será apresentado o modelo de média-variância, suas deficiências e técnicas propostas posteriormente visando aprimorar diferentes aspectos deste modelo.

2.1 Modelo Média-Variância

2.1.1 História da Teoria Moderna do Portfólio

A Teoria Moderna do Portfólio teve seu surgimento com o artigo publicado por Markowitz em 1952 [1], sendo mais detalhada posteriormente em seu livro [2]. Estes trabalhos foram marcos, tendo contribuído para sua premiação em 1990 com o

Prêmio Nobel de Economia. Nestes ele descreveu o risco associado a um ativo como sendo a variância dos seus retornos ao longo do tempo. Desta forma, formulou o problema de seleção de portfólio como um problema da escolha da valor esperado do retorno e da variância deste portfólio. No seu primeiro artigo [1] Markowitz rejeitou a ideia de que um investidor deveria se restringir a buscar um investimento que maximizasse seu retorno descontado. Assim como também rejeitou a lei dos grandes números na diversificação do portfólio, uma vez que os ativos são correlacionados demais para se assumir que seria possível anular o risco através de uma diversificação arbitrariamente grande.

Para Markowitz, um portfólio deveria ser escolhido baseando-se nos fatos incontestáveis de que um investidor racional sempre vai buscar ter um maior retorno com o menor risco possível, porém, cada investidor teria um determinado grau de tolerância a risco. Uma das contribuições foi definir o conceito de portfólio eficiente. Para ele, um portfólio é eficiente quando ele possui a menor variância para um determinado valor esperado de retorno. Ou seja, qualquer variação da proporção em que o capital está alocado só conseguirá reduzir a variância ao aceitar uma taxa de retorno menor. O mesmo vale para aquele portfólio que possui o maior valor esperado de retorno para uma determinada variância.

Porém, pode-se dizer que a maior contribuição do seu trabalho foi concluir que para definir a alocação ótima em um portfólio não basta observar o comportamento de cada ativo individualmente. É preciso que se avalie como os retornos dos ativos variam conjuntamente, pois ao levar isso em consideração é possível construir um portfólio com o mesmo valor esperado, porém, com menor risco do que se essa informação não fosse utilizada.

2.1.2 Problema de Otimização dos Pesos

Markowitz resume o problema de seleção de portfólio em um problema da escolha de um balanço entre retorno e risco em um portfólio. A medida de risco adotada por ele é a variância dos retornos. Nesta seção estes conceitos são definidos, assim como o problema de otimização utilizado para encontrar os portfólios eficientes. A notação utilizada difere da utilizada originalmente. Isto foi feito a fim de explicitar que os retornos são modelados como um processo estocástico. Cada amostra de uma série temporal de retornos é tratada como uma realização de uma variável aleatória, pertencente a um processo estocástico de retornos.

Dado um portfólio com N ativos, usa-se i como índice para referenciar um destes ativos e t para se referir a um determinado instante de tempo. Desta forma tem-se

$$i = 1, \dots, N$$

$$t = 1, \dots, T$$

onde T é o total de amostras disponíveis.

Pode-se então definir o retorno de um ativo i no instante de tempo t em função do preço como

$$r_i(t) = \frac{p_i(t) - p_i(t-1)}{p_i(t-1)}$$

onde $p_i(t)$ é o preço do ativo i no instante t .

Este retorno no instante de tempo t pode ser considerado como uma realização da variável aleatória $R_i(t)$ pertencente a um processo estocástico de retornos definido como

$$\{R_i(t) : t \in [1, T]\}$$

Desta forma, finalmente define-se o valor esperado do retorno e a variância de um ativo i no instante de tempo t como

$$\mu_i(t) = \mathbb{E}[R_i(t)]$$

$$\sigma_i^2(t) = \mathbb{E}[(R_i(t) - \mu_i(t))^2]$$

No modelo de média-variância o retorno do portfólio é então definido como

$$R_p(t) = \sum_{i=1}^N w_i R_i(t)$$

sendo w_i a quantidade relativa investida no ativo i , de forma que

$$\sum_{i=1}^N w_i = 1$$

Agora é possível definir tanto o valor esperado do portfólio quanto sua variância em termos do valor esperado e variância dos retornos dos ativos que o compõem. Seu valor esperado será

$$\mu_p(t) = \sum_{i=1}^N w_i \mu_i(t) \tag{2.1}$$

$$\sigma_p^2 = \sum_i \sum_j w_i w_j cov_{i,j}(t) \tag{2.2}$$

onde $cov_{i,j}(t)$ é a covariância entre dois ativos i e j no instante t .

A Equação 2.1 e a Equação 2.2 podem ser reescritas de forma matricial como

$$\mu_p(t) = \mathbf{w}^T \boldsymbol{\mu}_t \tag{2.3}$$

$$\sigma_p^2(t) = \mathbf{w}^T \Sigma \mathbf{w} \quad (2.4)$$

Destá forma o problema de seleção de portfólios pode ser modelado por meio de um problema de Programação Linear. Neste a variância será minimizada para um determinado valor esperado de retornos.

$$\begin{aligned} &\text{Minimize} && \mathbf{w}^T \Sigma \mathbf{w} \\ &\text{sujeito a} && \mathbf{w}^T \boldsymbol{\mu}_t = \mu_p(t), \\ &&& \mathbf{w}^T \mathbf{1}_N = 1 \end{aligned} \quad (2.5)$$

onde $\mu_p(t)$ é retorno alvo estabelecido e $\mathbf{1}_N$ é um vetor de comprimento N com todos os elementos iguais a 1. Este problema é facilmente resolvido utilizando multiplicadores de Lagrange. Primeiramente o problema será reescrito como uma função de Lagrange.

$$\mathcal{L}(\mathbf{w}, \lambda_1, \lambda_2) = \frac{1}{2} \mathbf{w}^T \Sigma \mathbf{w} + \lambda_1 (1 - \mathbf{w}^T \mathbf{1}_N) + \lambda_2 (\mu_p(t) - \mathbf{w}^T \boldsymbol{\mu}_t)$$

A seguir calcula-se o gradiente da função de Lagrange e iguala-se a 0.

$$\nabla_{\mathbf{w}, \lambda_1, \lambda_2} \mathcal{L} = 0 \quad \Leftrightarrow \quad \begin{cases} \Sigma \mathbf{w} - \lambda_1 \mathbf{1}_N - \lambda_2 \boldsymbol{\mu}_t = 0 \\ 1 - \mathbf{w}^T \mathbf{1}_N = 0 \\ \mu_p(t) - \mathbf{w}^T \boldsymbol{\mu}_t = 0 \end{cases} \quad (2.6)$$

Resolvendo \mathbf{w}^* em função de λ_1 e λ_2 o vetor de pesos do portfólio admite uma solução na forma

$$\mathbf{w}^* = \lambda_1 \Sigma^{-1} \mathbf{1}_N + \lambda_2 \Sigma^{-1} \boldsymbol{\mu}_t \quad (2.7)$$

A fim de determinar λ_1 e λ_2 aplica-se a Equação 2.7 às duas últimas componentes da Equação 2.6. A primeira é reescrita como

$$\begin{aligned} \mathbf{w}^T \boldsymbol{\mu}_t &= \mu_p(t) \\ (\lambda_1 \Sigma^{-1} \mathbf{1}_N + \lambda_2 \Sigma^{-1} \boldsymbol{\mu}_t)^T \boldsymbol{\mu}_t &= \mu_p(t) \\ \lambda_1 \boldsymbol{\mu}_t^T \Sigma^{-1} \mathbf{1}_N + \lambda_2 \boldsymbol{\mu}_t^T \Sigma^{-1} \boldsymbol{\mu}_t \boldsymbol{\mu}_t &= \mu_p(t) \end{aligned} \quad (2.8)$$

e a segunda, como

$$\begin{aligned} \mathbf{w}^T \mathbf{1}_N &= 1 \\ (\lambda_1 \Sigma^{-1} - \mathbf{1}_N + \lambda_2 \Sigma^{-1} \boldsymbol{\mu}_t)^T \mathbf{1}_N &= 1 \\ \lambda_1 \mathbf{1}_N^T \Sigma^{-1} \mathbf{1}_N + \lambda_2 \mathbf{1}_N^T \Sigma^{-1} \boldsymbol{\mu}_t \boldsymbol{\mu}_t &= \mu_p(t) \end{aligned} \quad (2.9)$$

Define-se então

$$a \equiv \mathbf{1}_N^T \Sigma^{-1} \mathbf{1}_N$$

$$b \equiv \mathbf{1}_N^T \Sigma^{-1} \boldsymbol{\mu}_t$$

$$c \equiv \boldsymbol{\mu}_t^T \Sigma^{-1} \boldsymbol{\mu}_t$$

de forma que pode-se então reescrever as Equações 2.8 e 2.9 em termos de a , b e c .

$$\lambda_1 a + \lambda_2 b = 1 \quad (2.10)$$

$$\lambda_1 b + \lambda_2 c = \mu_p(t) \quad (2.11)$$

Resolvendo as Equações 2.10 e 2.11 para λ_1 e λ_2 tem-se

$$\lambda_1 = \frac{c - b\mu_p(t)}{\Delta} \quad (2.12)$$

$$\lambda_2 = \frac{a\mu_p(t) - b}{\Delta} \quad (2.13)$$

onde $\Delta = ac - b^2$. O que é válido, pois assume-se que $\boldsymbol{\mu}_t \neq h\mathbf{1}_N$ e que Σ^{-1} existe. Uma vez que Σ é positiva definida, então $a > 0$, $c > 0$. Em virtude da desigualdade de Cauchy-Schwarz tem-se que $\Delta > 0$.

Portanto pode-se reescrever a Equação 2.7 como

$$\mathbf{w}^* = \frac{(c - b\mu_p(t))\Sigma^{-1}\mathbf{1}_N + (a\mu_p(t) - b)\Sigma^{-1}\boldsymbol{\mu}_t}{ac - b^2} \quad (2.14)$$

o que dá a combinação ótima de pesos para um determinado universo de ativos, uma vez fixado um valor esperado de retorno para o portfólio.

2.1.3 Fronteira Eficiente

Também é possível determinar a variância dos portfólios em função do valor esperado de retorno.

$$\sigma_p^2(t) = \mathbf{w}^{*T} \Sigma \mathbf{w}^*$$

$$\sigma_p^2(t) = \mathbf{w}^{*T} \Sigma (\lambda_1 \Sigma^{-1} \mathbf{1}_N + \lambda_2 \Sigma^{-1} \boldsymbol{\mu}_t)$$

$$\sigma_p^2(t) = \lambda_1 \mathbf{w}^{*T} \Sigma \Sigma^{-1} \mathbf{1}_N + \lambda_2 \mathbf{w}^{*T} \Sigma \Sigma^{-1} \boldsymbol{\mu}_t$$

$$\sigma_p^2(t) = \lambda_1 \mathbf{w}^{*T} \mathbf{1}_N + \lambda_2 \mathbf{w}^{*T} \boldsymbol{\mu}_t$$

de forma que ao aplicar as restrições da Equação 2.5 obtém-se

$$\sigma_p^2(t) = \lambda_1 + \lambda_2 \mu_p(t) \quad (2.15)$$

assim sendo, basta então substituir as Equações 2.12 e 2.13 na Equação 2.15 para obter

$$\sigma_p^2(t) = \frac{a\mu_p(t)^2 - 2b\mu_p(t) + c}{\Delta} \quad (2.16)$$

Fica evidente então, que a variância de um portfólio apresenta uma relação quadrática com o retorno esperado, como explicitado pela Figura 2.4. Pode-se observar que o ponto $(\frac{1}{a}, \frac{b}{a})$ marca o portfólio de mínima variância. Acima dele todos os pontos dizem respeito a um portfólio eficiente. Para qualquer um destes não existe outra combinação entre os ativos que resulte em um retorno esperado maior com o mesma variância, nem uma variância menor com o mesmo retorno. Ao conjunto desses portfólios eficientes dá-se o nome de Fronteira Eficiente.

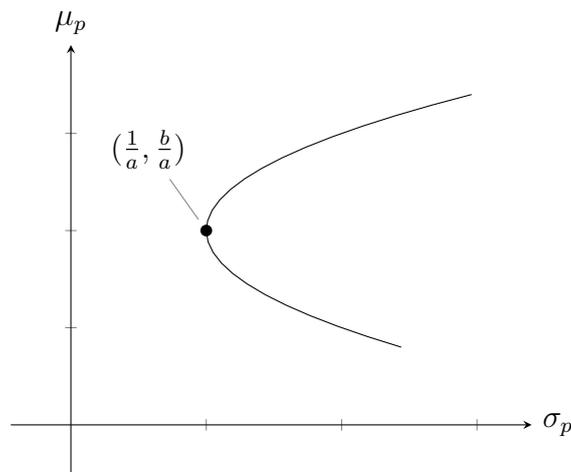


Figura 2.1: A relação entre o valor esperado de retornos e o risco, representado pelo desvio padrão. Vale destacar que o eixo vertical é comumente utilizado na literatura, para representar o retorno esperado, assim como normalmente utiliza-se o desvio padrão no eixo horizontal.

Para exemplificar pode-se considerar um universo específico de ativos, cada qual com um determinado valor esperado de retorno e uma variância específica. Para simplificar, considera-se que não há correlação entre eles. A Figura 2.2 mostra o valor esperado e variância de cada ativo individualmente, assim como o de diferentes combinações destes ativos em um portfólio. É evidente que todas as combinações estão contidas na região abaixo da fronteira eficiente. É também claro que é possível obter um portfólio com variância menor do que a do ativo de menor variância, com um valor esperado inclusive mais alto.

O exemplo da Figura 2.2 considerou ativos descorrelacionados. Porém, como mostra a Equação 2.2, a covariância entre os ativos que compõem o portfólio apresentam influência na variância deste. Esta influência fica evidente no exemplo retratado na Figura 2.3. Nela, é representado um universo de dois ativos, assim como as fronteiras eficientes para diferentes valores de correlação entre eles.

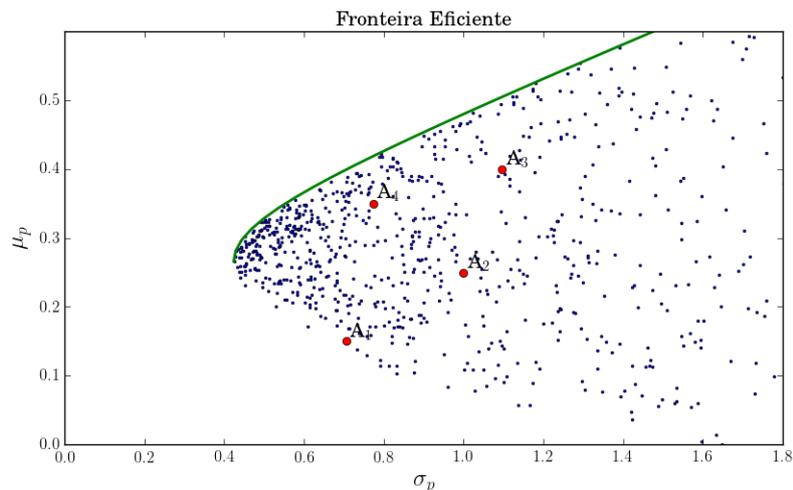


Figura 2.2: Fronteira eficiente em um universo de quatro ativos. São plotados em azul portfólios com pesos diferentes, em vermelho os quatro ativos individualmente e em verde a parábola que delimita a Fronteira Eficiente.

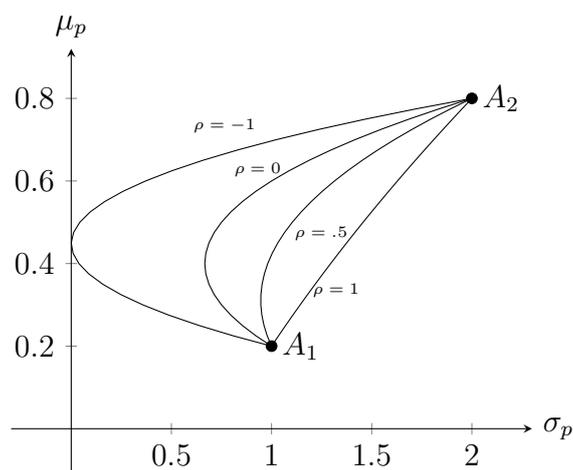


Figura 2.3: Fronteiras eficientes com diferentes valores de correlação entre os ativos em um universo de dois ativos.

2.1.4 Ativo Livre de Risco e o Portfólio Tangente

Um ativo livre de risco é um ativo que possui um valor esperado de retorno μ_0 associado a uma variância nula. Na prática é comumente considerado um ativo livre de risco títulos públicos de curto prazo de vencimento, como por exemplo títulos do tesouro americano. Um ativo livre de risco é, por definição, decorrelacionado de qualquer outro ativo, uma vez que possui uma variância nula.

Ao se considerar a possibilidade de incluir um ativo livre de risco em um portfólio passa-se a ter uma nova fronteira eficiente. Esta é agora a linha que passa pelo ponto referente ao ativo livre de risco e tangencia a fronteira eficiente antiga, passando pelo

que é chamado de Portfólio Tangente. Qualquer ponto nessa linha passa a ser um portfólio eficiente e é composto de uma combinação entre o ativo livre de risco e o portfólio tangente. No caso especial dos pontos à direita do portfólio tangente eles se referem ao caso em que se faz alavancagem no portfólio eficiente em relação ao ativo livre de risco. Ou seja, pega-se dinheiro emprestado à taxa do ativo livre de risco e o aplica no portfólio tangente.

O portfólio tangente também é conhecido como Portfólio de Mercado e referenciado pela letra M . Isso porque é aquele portfólio que seria adquirido em alguma proporção por qualquer investidor racional, independente do seu nível de aversão a risco. Investidores mais aversos a risco terão uma parcela menor investida no portfólio tangente e uma maior investida no ativo livre de risco, e o oposto para investidores menos aversos ao risco. Ainda assim, todos teriam uma parcela investida no portfólio tangente.

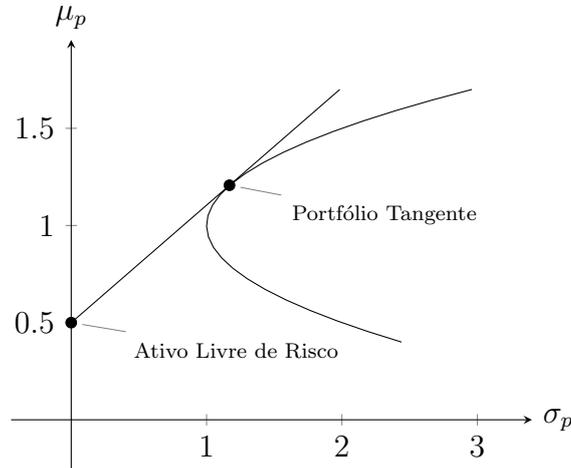


Figura 2.4: Fronteira eficiente considerando um ativo livre de risco. A reta que passa pelo portfólio tangente e pelo ativo livre de risco se torna a nova fronteira eficiente.

Para encontrar o portfólio tangente, deve-se reformular o problema anterior de forma que $\mathbf{w}^T \mathbf{1}_N$ é investido em ativos de risco e $1 - \mathbf{w}^T \mathbf{1}_N$ é investido no ativo livre de risco.

$$\begin{aligned} & \text{Minimize} && \mathbf{w}^T \Sigma \mathbf{w} \\ & \text{sujeito a} && \mathbf{w}^T \boldsymbol{\mu}_t + (1 - \mathbf{w}^T \mathbf{1}_N) \mu_0 = \mu_p(t), \end{aligned} \tag{2.17}$$

Define-se a função de Lagrange como

$$\mathcal{L}(\mathbf{w}, \lambda) = \frac{1}{2} \mathbf{w}^T \Sigma \mathbf{w} + \lambda [(\mu_p(t) - \mu_0) - \mathbf{w}^T (\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)]$$

A seguir calcula-se o gradiente da função de Lagrange e iguala-se a 0.

$$\nabla_{\mathbf{w}, \lambda} \mathcal{L} = 0 \quad \Leftrightarrow \quad \begin{cases} \boldsymbol{\Sigma} \mathbf{w} - \lambda(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0) = 0 \\ (\mu_p(t) - \mu_0) - \mathbf{w}^T(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0) = 0 \end{cases} \quad (2.18)$$

Resolvendo \mathbf{w}^* em função de λ_2 e λ_2 o vetor de pesos do portfólio admite uma solução na forma

$$\mathbf{w}^* = \lambda \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0) \quad (2.19)$$

A fim de determinar λ aplica-se a Equação 2.19 à última componente da Equação 2.18. Tem-se então

$$\begin{aligned} (\mu_p(t) - \mu_0) - [\lambda \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)]^T(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0) &= 0 \\ \lambda(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0) &= (\mu_p(t) - \mu_0) \\ \lambda &= \frac{\mu_p(t) - \mu_0}{(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)} \end{aligned} \quad (2.20)$$

De forma que os pesos para o portfólio tangente são dados substituindo a Equação 2.20 na Equação 2.19

$$\mathbf{w}^* = \frac{(\mu_p(t) - \mu_0) \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)}{(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)} \quad (2.21)$$

E a função que relaciona a variância e a média dá-se por

$$\begin{aligned} \boldsymbol{\Sigma} \mathbf{w}^* &= \lambda(\boldsymbol{\mu}_t - \mathbf{1}_N) \\ \mathbf{w}^{*T} &= \lambda(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)^T \boldsymbol{\Sigma}^{-1} \\ \sigma_p^2(t) &= \mathbf{w}^{*T} \boldsymbol{\Sigma} \mathbf{w}^* = \lambda^2(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0) \\ \sigma_p^2(t) &= \frac{(\mu_p(t) - \mu_0)^2}{(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N \mu_0)} \end{aligned} \quad (2.22)$$

Reorganizando o denominador da Equação 2.22 em termos de a , b e c , obtém-se

$$\sigma_p^2(t) = \frac{(\mu_p(t) - \mu_0)^2}{a\mu_0^2 - 2b\mu_0 + c} \quad (2.23)$$

que delimita duas retas tangentes a parábola da fronteira eficiente.

$$\mu_p(t) = \mu_0 \pm \sigma_p(t) \sqrt{a\mu_0^2 - 2b\mu_0 + c}$$

Lembrando que $a\mu_0^2 - 2b\mu_0 + c > 0$, pois, como já foi mostrado, $\Delta = ac - b^2 > 0$.

Apenas a reta superior é de interesse e ela representa a nova fronteira eficiente no modelo média-variância com um ativo livre de risco. Portanto

$$\mu_p(t) = \mu_0 + \sigma_p(t) \sqrt{a\mu_0^2 - 2b\mu_0 + c} \quad (2.24)$$

Por fim, o portfólio tangente M pode ser representado pelo ponto (σ_M, μ_M) , que pode ser obtido resolvendo simultaneamente a Equação 2.16 e a Equação 2.24, de forma a obter

$$\mathbf{w}_M^* = \frac{\Sigma^{-1}(\boldsymbol{\mu}_t - \mathbf{1}_N\mu_0)}{b - a\mu_0} \quad (2.25)$$

$$\mu_M(t) = \frac{c - b\mu_0}{b - a\mu_0} \quad (2.26)$$

$$\sigma_M^2(t) = \frac{a\mu_0^2 - 2b\mu_0 + c}{(b - a\mu_0)^2} \quad (2.27)$$

No exemplo mostrado anteriormente com o universo de quatro ativos decorrelacionados pode-se observar na Figura 2.5 a nova fronteira eficiente ao incluir um ativo livre de risco.

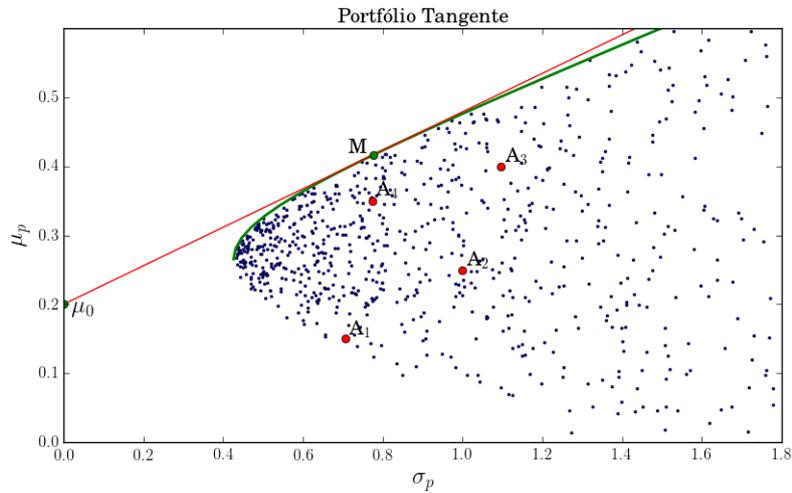


Figura 2.5: Fronteira eficiente em um universo de quatro ativos e um ativo livre de risco. São representados em azul portfólios com pesos diferentes, em vermelho os quatro ativos individualmente e em verde o ativo livre de risco e o portfólio tangente. A curva em verde representa a fronteira eficiente caso não houvesse um ativo livre de risco e a reta vermelha é a fronteira eficiente ao considerar um ativo livre de risco.

2.1.5 Deficiências do Modelo Média-Variância

Como foi dito previamente, todo investidor racional buscaria dividir seu investimento entre um ativo livre de risco e o portfólio tangente. Na prática, porém, algumas

deficiências podem ser apontadas com relação ao modelo de média-variância, que tornam impraticável encontrar o portfólio tangente.

Para calcular os pesos do portfólio tangente utilizando a Equação 2.25 é necessário estimar o valor esperado dos retornos de cada ativo para o próximo instante de tempo, assim como suas variâncias e covariâncias. Este número cresce quadraticamente com o tamanho do universo de ativos no portfólio, sendo igual a $\frac{N(N+1)}{2} + N$, onde N é o tamanho do universo de ativos no portfólio. Tradicionalmente, conforme sugerido por Markowitz em 1952 [1] estes parâmetros podem ser calculados a partir da média amostral e da covariância amostral, dadas por

$$\hat{\mu} = \frac{1}{T} \sum_{t=0}^T r_t$$

$$\hat{\Sigma} = \frac{1}{T-1} \sum_{t=0}^T (r_t - \hat{\mu})(r_t - \hat{\mu})^T$$

O uso da média e covariância amostrais utilizando toda a série de retornos estaria assumindo que o processo estocástico dos retornos dos ativos é um processo estacionário e ergótico. Assumir isto é uma forte simplificação, visto que há evidências mostrando resultados positivos ao considerar que séries temporais de ativos financeiros são não-estacionárias [4, 5]. O que é comumente utilizado na prática é calcular a média e covariância amostrais utilizando uma janela com amostras próximas. Esta abordagem, por sua vez, também incorre em dificuldades para estimar os parâmetros. Isto porque, uma vez que quanto menos amostras são utilizadas para o cálculo da matriz de correlação (especialmente no caso de um universo com muitos ativos), mais imprecisa sua estimativa se torna.

Outro problema está no uso da variância como medida de risco. Isto porque ela considera equivalentes os retornos acima e abaixo do valor esperado. Do ponto de vista prático, retornos acima do valor esperado não são considerados negativos para um investidor.

2.2 Propostas para a Estimativa dos Parâmetros

Serão apresentadas, nesta seção, propostas para mitigar estas deficiências, assim como as próprias deficiências destas propostas. Inicialmente serão apresentadas diversas propostas de modelos matemáticos e técnicas utilizadas para se obter boas estimativas. Em seguida, será abordada a aplicação de técnicas de aprendizado de máquina na modelagem do retorno de ativos, incluindo a aplicação das mesmas no problema de seleção de portfólios.

2.2.1 Modelo de Índice Único

A modelagem estatística do retorno de ativos se popularizou a partir dos trabalhos de William Sharpe [10, 11], pelo qual o mesmo recebeu o Prêmio Nobel de Economia de 1990, juntamente com Markowitz. Em seu trabalho Sharpe desenvolveu o que ficou conhecido como Modelo de Índice Único (Single-Index Model - SIM). Este modelo assume que os retornos dos diferentes ativos podem ser modelados apenas por meio de sua interação com o retorno do mercado. Este é normalmente representado por algum índice de mercado como, por exemplo, o S&P500. O SIM modela o retorno em excesso em relação a um ativo livre de risco, como

$$r_i(t) - \mu_0 = \alpha_i + \beta_i(r_m(t) - \mu_0) + \epsilon_i(t)$$
$$r_m(t) = \alpha_m + \epsilon_m$$

onde $r_m(t)$ é o retorno do mercado em um instante de tempo t ; α é a medida do retorno anormal, ou seja, o excesso de retorno relação ao mercado; β é a taxa de sensibilidade ao mercado; $\epsilon_i(t)$ é o retorno residual de um ativo, que é modelado como uma variável aleatória, comumente gaussiana e independente.

Os parâmetros para a aplicação no modelo de média variância podem ser facilmente extraídos a partir destas duas equações, sendo

$$\mu_i(t) = \alpha_i + \beta_i \alpha_m \sigma_i^2(t) = \beta_i^2 \sigma_m^2 + \sigma_{\epsilon_i(t)} \text{cov}_{i,j}(t) = \beta_i \beta_j \sigma_m^2, \forall i \neq j$$

Devido a esta estrutura teórica que o modelo impõe ao problema, a quantidade de parâmetros a ser calculada reduz drasticamente, uma vez que a matriz de correlação é simplificada. Isto tornou o modelo muito mais simples para a compreensão de analistas e não só aumentou a acurácia na seleção de portfólio como também garantiu que a acurácia do modelo de mercado na estimação da matriz de covariância fosse maior do que ao estimá-la diretamente [12].

2.2.2 Modelos de Multi-Fator

Pouco tempo depois começaram a ser desenvolvidos modelos multi-índices, também conhecidos como multi-fator. Eles seguem a intuição de que os retornos de um dado ativo podem, além de se relacionar com um índice único de mercado, se relacionar com múltiplos outros fatores.

Genericamente um modelo multi-fator pode ser descrito por

$$r_i(t) - \mu_0 = \alpha_i + \beta_i(r_m(t) - \mu_0) + \epsilon_i(t) + \sum_{f=1}^F \beta_{i,f} \mathcal{F}_f(t) + \epsilon_i$$

onde F é o total de fatores no modelo, \mathcal{F}_f é o valor de um determinado fator num instante de tempo t , $\beta_{i,f}$ é a taxa de sensibilidade de um ativo a um determinado fator.

Uma grande motivação foi ao observar que mesmo depois de se remover a covariância com o índice de mercado ainda sobra covariância entre ativos da mesma indústria [13]. Cohen e Pogue desenvolveram em 1967 [14] um modelo para o retorno em excesso de um ativo, usando fatores referentes ao retorno em excesso de diferentes indústrias. Por sua vez, modelam o retorno em excesso de uma indústria como uma relação linear com o retorno em excesso do mercado ($ar_m(t) + b$).

Fama e French desenvolveram em 1992 [15] um modelo multi-fator, que se tornou muito popular, usando três fatores: o risco do mercado, a sobre performance de empresas pequenas versus grandes e a sobre performance de empresas com alto price-to-book ratio versus baixo. O price-to-book é, por sua vez, a razão do preço de uma empresa e seu valor patrimonial, sendo usado como métrica para buscar por empresas subvalorizadas. O modelo de três fatores pode ser descrito por

$$r_i(t) - \mu_0 = \beta_i(r_m(t) - \mu_0) + b_s \cdot SMB + b_v \cdot HML + \alpha_i$$

onde SMB (Small Minus Big) é o valor do excesso de retorno de small caps (ativos de pequeno porte) em relação a big caps, HML (High Minus Low) é o excesso de retorno de ativos de alto valor em relação a ativos em crescimento, levando em consideração o price-to-book. Por fim, b_s e b_v são coeficientes a serem calculados via regressão linear, dados SMB e HML [16].

Diversos outros modelos multi-fator foram desenvolvidos no decorrer do tempo e continuam sendo desenvolvidos na atualidade. Fama e French desenvolveram novos modelos acrescentando novos fatores no seu modelo clássico [17, 18]. Pelger [19] e Aït-Sahalia [20] utilizaram PCA para determinar fatores para seus respectivos modelos. Já Harvey [21] catalogou 316 anomalias propostas como possíveis fatores.

Apesar de apresentarem modelos mais complexos do que o modelo de único fator os modelos de múltiplos fatores ainda continuam baseando-se em simplificações do comportamento das séries de retornos dos ativos de forma a conseguir modelar este comportamento de forma explícita. Por um lado isso facilita na compreensão de quais os fatores mais relevantes para um determinado movimento no preço de um ativo, mas por outro limita o grau de complexidade e, portanto, de precisão que um modelo é capaz de atingir.

2.2.3 Métodos de Encolhimento

Como já foi descrito previamente, um dos problemas no modelo de média-variância clássico está na estimação do valor esperado e na matriz de covariâncias para o

próximo instante de tempo. Como a série de retornos dos ativos não provêm de um processo estacionário não é viável a utilização de todas as amostras na construção da matriz de covariância amostrais, sendo comumente utilizada uma janela temporal destas amostras. Quando o número de amostras é reduzido ou quando o número de ativos no portfólio é muito elevado, as estimativas amostrais, especialmente da matriz de covariância tornam-se muito imprecisas. Isto levou ao uso de técnicas de encolhimento, a fim de poder obter melhores estimativas da matriz de covariância do que seria possível simplesmente através do uso da covariância amostral.

O conceito de encolhimento foi proposto por Stein et al. em 1956 [22]. Stein demonstrou que, caso admita-se um estimador viesado, é possível obter um estimador com valores de erro médio quadrático (Mean Squared Error - MSE) inferior ao MSE de estimadores de máxima verossimilhança, ou seja, estimadores sem viés. Esses estimadores compõem a classe do que denomina-se estimadores de encolhimento (shrinkage). Um exemplo simples deste efeito é a estimação da variância de uma população. Para uma população de n amostras o uso de $n + 1$ como divisor garante um estimador sem viés, porém outros divisores podem apresentar resultados de MSE inferiores em troca de adicionar viés ao estimador.

As primeiras aplicações de estimadores de encolhimento aplicados a seleção de portfólio datam de 1986 [23] [24], mas se popularizaram apenas após o trabalho de Ledoit et al. [25] [26] [27], a partir de 2003, com uma série de propostas de estimadores de encolhimento para a matriz de covariância de um portfólio de ativos.

Em sua primeira proposta [25] utiliza-se uma média ponderada da matriz de correlação amostral e da matriz de covariância do modelo de índice único de Sharpe como estimador de encolhimento, descrito pela Equação 2.28.

$$\hat{\Sigma}_{shrink} = \hat{\lambda}^* F + (1 - \hat{\lambda}^*) \hat{\Sigma} \quad (2.28)$$

onde $\hat{\Sigma}$ é a matriz de covariância amostral, F é o alvo do encolhimento, neste caso a matriz de covariância calculada a partir do modelo de índice único, e $\hat{\lambda}^*$ é a estimação da constante de encolhimento ótima (λ^*).

Em seguida Ledoit et al. fez uso de um estimador de encolhimento com um alvo diferente, uma matriz de correlação constante [26]. Esta matriz é calculada de maneira que todos os pares de correlação entre os ativos sejam constantes e iguais a média da correlação na matriz de correlação amostral. Nos resultados empíricos este modelo apresentou resultados similares ao modelo que utiliza a matriz de covariância de índice único com o diferencial de ter uma implementação mais simples. Este modelo também pode ser descrito pelas Equações Equação 2.28 com a diferença que a matriz F agora representa a matriz de correlações constantes.

O terceiro modelo proposto por Ledoit et al. [27] utiliza como alvo do encolhi-

mento, F , a matriz diagonal com a média das variâncias dos ativos nas diagonais e zeros nas demais posições da matriz. Esse modelo apresenta como vantagem ser bem condicionado e ser facilmente calculado e de fácil compreensão.

A método de Encolhimento por Aproximação Oráculo (Oracle Approximating Shrinkage - OAS) proposto por Chen et al. [28] é outra técnica bastante popular devido ao fato de apresentar um menor MSE do que o do modelo de Ledoit et al. [27], para o caso de distribuições Gaussianas.

Apesar dos resultados superiores para o cálculo do valor esperado dos retornos e da matriz de covariância, todos os métodos apresentados continuam tendo diversas deficiências que podem influenciar na qualidade dos portfólios gerados. Modelos multi-fator modelam o processo estocástico dos retornos de forma a conseguir uma representação explícita, muitas vezes de fácil compreensão no sentido de quais fatores afetam o comportamento do retorno, atribuindo o comportamento dos retornos a um pequeno grupo de fatores. Estimadores de encolhimento, por sua vez, partem do pressuposto que as amostras temporais possuem a mesma distribuição estatística do que a da amostra futura. O trabalho de DeMiguel et al. [3] compara diversos modelos que buscam uma alocação de portfólio ótima e mostra que nenhum deles é consistentemente melhor do que uma alocação igualmente distribuída.

Tanto modelos multi-fator quanto estimadores de encolhimento buscam mitigar o problema causado pela necessidade de se estimar uma matriz de covariância e valor esperado em um portfólio de muitos ativos com um número restrito de amostras. Este número restrito de amostras ocorre uma vez que não é possível considerar os retornos dos ativos como um processo estacionário, de forma que não se pode utilizar todas as amostras da série para o cálculo. Isso leva a utilização apenas de uma janela de amostras próximas ao instante em questão. Este projeto busca propor um modelo que seja capaz de utilizar todas as amostras disponíveis na série temporal dos retornos a fim de conseguir estimar a distribuição estatística para os retornos no futuro.

Capítulo 3

Redes Adversárias Generativas Aplicadas a Séries Temporais

Neste capítulo serão apresentados os conceitos de Redes Neurais Artificiais utilizados neste projeto. Será inicialmente apresentado um breve histórico acerca de Redes Neurais e seu paralelo biológico. A seguir é abordada a aplicação de Redes Neurais, mais especificamente, Redes Neurais Recorrentes, em problemas envolvendo séries temporais. Por fim apresenta-se o conceito de Redes Generativas Adversárias e sua aplicação em séries temporais.

3.1 Redes Neurais Artificiais

O sistema nervoso humano é capaz de organizar e processar informações e desempenhar diversas tarefas altamente complexas, como reconhecimento de padrões e controle motor. Os neurônios, seu principal constituinte, são células eletricamente excitáveis [29] que se comunicam entre si por meio de variações na diferença de potencial (tensão) que possuem. O cérebro é o centro do sistema nervoso, possuindo na ordem de 86 bilhões de neurônios [30]

Redes Neurais Artificiais são uma classe de modelos computacionais que possuem uma inspiração nas redes neurais biológicas. Elas foram concebidas a partir da tentativa de emular neurônios biológicos e são treinadas com dados reais a fim de que sejam capazes de realizar uma gama de tarefas complexas.

3.1.1 Neurônio Artificial

Tomando como base os conceitos biológicos do neurônio, o primeiro modelo de um neurônio artificial foi proposto em 1943 por McCulloch e Pitts [31]. Na Figura 3.1, pode-se observar a representação gráfica do modelo, nele o neurônio possui diversas entradas binárias (x_N), fazendo o papel dos dendritos de um neurônio biológico.

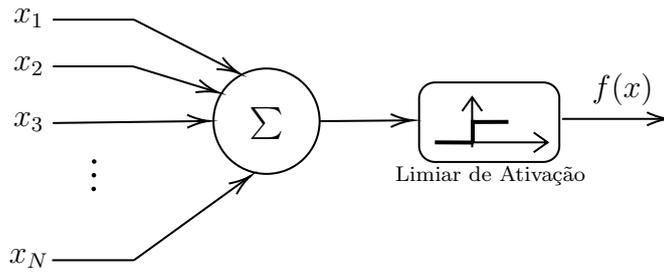


Figura 3.1: Modelo McCulloch-Pitts de um neurônio artificial

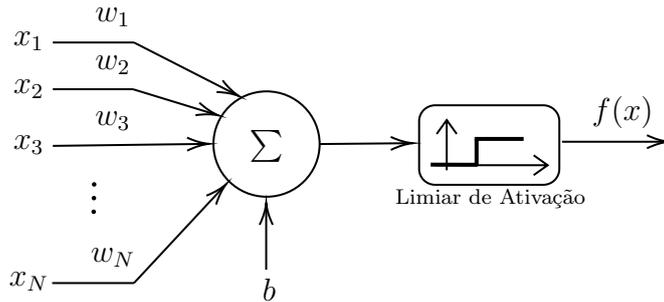


Figura 3.2: Modelo Perceptron de um neurônio artificial

Estas entradas são somadas e, caso o resultado seja superior a um determinado limiar, o neurônio fica ativo. A saída do neurônio, que faz o papel do axônio, retorna um valor binário dependendo se o neurônio está ou não ativo. O modelo de McCulloch-Pitts pode ser representado pela Equação 3.1.

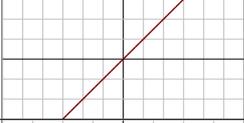
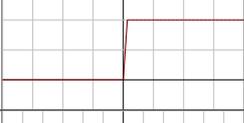
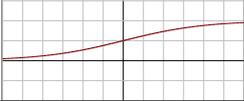
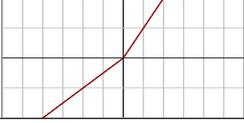
$$f(x) = \begin{cases} 1, & \text{se } \sum_{i=0}^N x_i > \theta \\ 0, & \text{caso contrário} \end{cases} \quad (3.1)$$

onde x é um vetor binário de tamanho N e θ é o limiar de ativação.

O modelo Perceptron [32], foi proposto como uma forma de superar algumas limitações do modelo de McCulloch-Pitts, sendo uma generalização do mesmo. Diferente do modelo de McCulloch-Pitts as entradas do Perceptron podem ser números reais. Estas são multiplicadas por pesos, que possibilitam que cada entrada possa contribuir mais ou menos com a ativação do neurônio. A representação do Perceptron pode ser vista na Figura 3.2.

Outro conceito novo no Perceptron é o viés (bias), que é uma entrada de valor fixo, independente das variáveis de entrada. Ele representa o quanto o neurônio precisa ser excitado para chegar a ativação. Os pesos e o bias do Perceptron podem ser alterados de forma que sua ativação aconteça de forma consistente com a classe de dados apresentada na sua entrada. A este processo, atribui-se a expressão treinamento. O Perceptron pode ser descrito pela Equação 3.2.

Tabela 3.1: Exemplos de funções de ativação comumente utilizadas. u representa a entrada da função de ativação, ou seja, $u = w \odot x + b$.

Nome	Gráfico	Equação
Linear		$f(u) = u$
Degrau		$\begin{cases} 1, & \text{se } u > 0 \\ 0, & \text{caso contrário} \end{cases}$
Sigmoid		$f(u) = \frac{1}{1 + e^{-u}}$
Tangente Hiperbólica		$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$
ReLU		$f(u) = \begin{cases} u, & \text{se } u \geq 0 \\ 0, & \text{caso contrário} \end{cases}$
Leaky ReLU		$f(\alpha, u) = \begin{cases} u, & \text{se } u > 0 \\ \alpha u, & \text{caso contrário} \end{cases}$

$$f(x) = \begin{cases} 1, & \text{se } w \odot x + b > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.2)$$

sendo x é o vetor de entradas do Perceptron. w é o vetor de pesos e b é o bias e \odot representa o produto de Hadamard, que é o produto elemento a elemento entre os vetores.

Apesar de originalmente o Perceptron utilizar um degrau como função de ativação, diversas outras funções de ativação foram propostas. Denota-se a função de ativação de um neurônio por σ . A Tabela 3.1 mostra algumas das funções de ativação mais comumente utilizadas.

Tanto o Perceptron quanto o modelo de McCulloch-Pitts podem ser interpretados como um hiperplano separando o espaço em dois, tornando-os classificadores lineares. Isto limita a aplicação de ambos a maior partes dos problemas reais. A solução para esta limitação está na combinação de neurônios de forma a conseguir separações mais complexas.

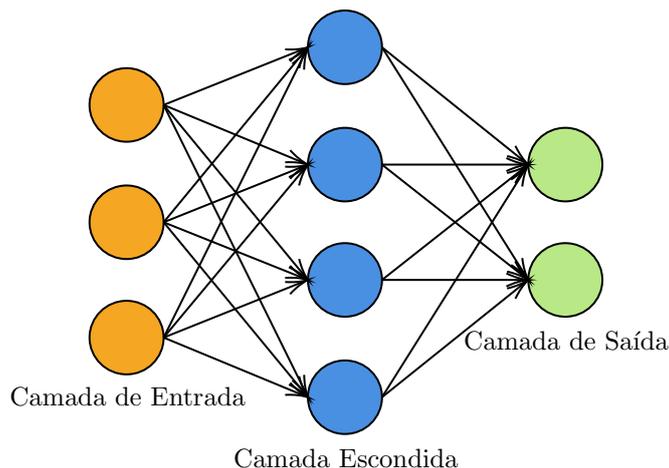


Figura 3.3: Modelo Perceptron Multicamada com 3 entradas, 4 neurônios na camada escondida e 2 neurônios na camada de saída

3.1.2 Backpropagation e o Perceptron Multicamadas

Devido a limitação dos modelos de McCulloch-Pitts e do Perceptron de apenas promover uma separação linear no espaço, redes neurais perderam popularidade. Este hiato durou até a década 80 quando o algoritmo de Backpropagation [33] [34] tornou viável o treinamento de redes neurais com múltiplos Perceptrons conectados em múltiplas camadas, conhecidas como Perceptron Multicamadas (Multi Layer Perceptron - MLP). Redes neurais MLP, segundo o Teorema da Aproximação Universal [35], possibilitam a representação de quaisquer funções em \mathbb{R}^n , com apenas uma camada escondida.

Uma rede MLP, representada na Figura 3.3, consiste em pelo menos 3 camadas de nós: uma camada de entrada; uma ou mais camadas escondidas e uma camada de saída. Com exceção dos nós da camada de entrada, todos os demais nós representam um neurônio. Em uma rede MLP, a saída de cada neurônio em uma camada serve como entrada para todos os neurônios da camada seguinte. Os pesos de cada neurônio são alterados a partir do algoritmo de backpropagation, de forma a reduzir uma função custo C definida, como por exemplo o MSE entre a saída atual e a saída desejada.

Pode-se representar os neurônios em uma rede MLP pela seguinte equação:

$$a_j^l = \sigma \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right) \quad (3.3)$$

onde a_j^l é a ativação no neurônio j da camada l , que se dá pela soma das ativações da camada anterior ($l - 1$), ponderadas pelos pesos da camada l acrescido do bias da camada l .

Pode-se simplificar a equação, expressando-a em função de z , que representa a

entrada da função de ativação, ou seja, a soma das entradas ponderadas pelos pesos com o bias. Esta representação será útil mais adiante.

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l \quad (3.4)$$

$$a_j^l = \sigma(z_j^l) \quad (3.5)$$

A origem do backpropagation tem suas raízes nas técnicas de otimização em Teoria de Controle usando gradiente [36]. Ela foi formalizada na dissertação de Paul Werbos em 1974 [33] e popularizada pelo trabalho de Rumelhart, Hinton e Williams em 1988 [34].

O algoritmo de backpropagation é utilizado para que se possa treinar uma rede neural MLP utilizando um algoritmo de otimização de Gradiente Descendente. Algoritmos de gradiente descendente buscam minimizar uma função custo através de iterativas atualizações nos parâmetros a serem otimizados. Estas atualizações são proporcionais ao gradiente da função custo em relação aos parâmetros.

Em redes neurais, os parâmetros a serem atualizados são os pesos e os bias de cada neurônio. Em redes com múltiplas camadas escondidas o algoritmo de backpropagation busca determinar como o encadeamento de camadas escondidas afeta a função custo. Para tal, faz-se uso da Regra da Cadeia de forma a conseguir calcular a derivada parcial da função custo C em relação a cada peso ($\frac{\partial C}{\partial w_{jk}^l}$) e bias ($\frac{\partial C}{\partial b_j^l}$). Sendo w_{jk}^l o peso k do neurônio j da camada l e b_j^l o bias do neurônio j da camada l .

Para o cálculo dessas derivadas parciais primeiro defini-se δ_j^l , que representa o quanto mudanças na entrada da função de ativação do neurônio j da camada l , afetam a função custo. Ou seja o quanto aquele neurônio contribui para o erro. Pode ser informalmente chamado de erro do neurônio j da camada l .

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l} \quad (3.6)$$

A fim de calcular o valor δ_j^l pode-se aplicar a regra da cadeia. Desta forma tem-se:

$$\delta_j^L = \sum_k \frac{\partial C}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} \quad (3.7)$$

Substituindo a Equação 3.6 na Equação 3.7, tem-se:

$$\delta_j^L = \sum_k \frac{\partial z_k^{l+1}}{\partial z_k^l} \delta_j^{l+1} \quad (3.8)$$

Usando a Equação 3.4 representa-se z_k^{l+1} por:

$$z_k^{l+1} = \sum_j w_{kj}^{l+1} a_j^l + b_k^{l+1} = \sum_j w_{kj}^{l+1} \sigma(z_j^l) + b_k^{l+1} \quad (3.9)$$

Desta forma, ao diferenciar em relação a z_k^l tem-se:

$$\frac{\partial z_k^{l+1}}{\partial z_k^l} = w_{kj}^{l+1} \sigma'(z_j^l) \quad (3.10)$$

Pode-se finalmente substituir a Equação 3.10 na Equação 3.8 de forma a obter:

$$\delta_j^L = \sum_k w_{kj}^{l+1} \delta_j^{l+1} \sigma'(z_j^l) \quad (3.11)$$

Reescrevendo a Equação 3.11 em forma matricial chega-se na seguinte equação:

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^L) \quad (3.12)$$

Pode-se observar que o erro em um dado neurônio só pode ser calculado uma vez que se possui o erro de cada neurônio na camada seguinte. Desta forma, basta calcular o erro dos neurônios da última camada L e retropropagá-lo, de forma a obter todos os erros. O erro da última camada também pode ser calculado usando a regra da cadeia da seguinte forma.

$$\delta_j^L = \sum_k \frac{\partial C}{\partial a_k^L} \frac{\partial a_k^L}{\partial z_k^L} \quad (3.13)$$

Como a_k^L/z_k^L desaparece para $k \neq j$, pode-se reescrever a Equação 3.13 como:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L} \quad (3.14)$$

Usando a definição de a_j^l da Equação 3.5, tem-se que:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L) \quad (3.15)$$

A Equação 3.15 pode ser reescrito em forma matricial como:

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \quad (3.16)$$

onde $\nabla_a C$ é o vetor cujos componentes são as derivadas parciais $\partial C/\partial a_j^L$

Agora é possível calcular o quanto mudanças em cada neurônio de uma rede neural afetam o resultado a função custo. Isso porém ainda não deixa claro como calcular os ajustes que devem ser feitos nos pesos e biases da rede. Para isso, deve-se mais uma vez utilizar a regra da cadeia.

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l} \quad (3.17)$$

Imediatamente pode-se observar que a primeira derivada parcial pode ser substituída pelo erro daquele neurônio e reescrita como:

$$\frac{\partial C}{\partial w_{jk}^l} = \sigma_j^l \frac{\partial z_j^l}{\partial w_{jk}^l} \quad (3.18)$$

Mais uma vez pode-se utilizar a definição de z_j^l da Equação 3.4. Ao diferenciá-la em relação a w_{jk}^l , tem-se:

$$\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1} \quad (3.19)$$

Substituindo a Equação 3.19 na Equação 3.18 chega-se a:

$$\frac{\partial C}{\partial w_{jk}^l} = \sigma_j^l a_k^{l-1} \quad (3.20)$$

Que em notação matricial torna-se:

$$\frac{\partial C}{\partial w_{jk}^l} = a^{l-1} (\sigma^l)^T \quad (3.21)$$

Por fim, para calcular a derivada parcial da função custo em relação a um bias basta aplicar mais uma vez a regra da cadeia:

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial b_j^l} \quad (3.22)$$

Novamente pode-se observar que a primeira derivada parcial pode ser substituída pelo erro do neurônio e reescrita como:

$$\frac{\partial C}{\partial b_j^l} = \sigma_j^l \frac{\partial z_j^l}{\partial b_j^l} \quad (3.23)$$

Derivando mais uma vez a Equação 3.4, desta vez é fácil notar que o somatório se tornará igual a zero ao ser diferenciado e o bias igual a um.

$$\frac{\partial z_j^l}{\partial b_j^l} = 1 \quad (3.24)$$

De forma que a Equação 3.23 pode ser reescrita como:

$$\frac{\partial C}{\partial b_j^l} = \sigma_j^l \quad (3.25)$$

Torna-se, então, possível calcular as derivadas parciais da função custo em relação

Tabela 3.2: Equações do Algoritmo de Backpropagation

1	$\delta^L = \nabla_a C \odot \sigma'(z^L)$
2	$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^L)$
3	$\frac{\partial C}{\partial w_{jk}^l} = a^{l-1} (\sigma^l)^T$
4	$\frac{\partial C}{\partial b_j^l} = \sigma_j^l$

aos pesos e aos biases através do erro em cada neurônio, previamente calculado.

O algoritmo de backpropagation pode ser resumido às Equações 3.12, 3.16, 3.21, 3.25 exibidas na Tabela 3.2. O algoritmo 1 representa as equações, calculando o gradiente da função custo. Com o valor do gradiente em mãos pode-se utilizar qualquer variante do algoritmo de gradiente descendente para alterar os parâmetros da rede neural a fim de reduzir o valor da função custo.

Algoritmo 1: Backpropagation

Entrada: Uma amostra x do conjunto de treinamento

Saída: Gradiente $\nabla_{w,b} C$ da função custo com relação aos pesos e biases da rede neural

$$a^1 = w^1 x + b$$

// Passa-se a entrada pela rede neural (forward pass)

para $l = 2$ **até** L **faça**

$$| z^l = w^l a^{l-1} + b^l$$

fim

// Calcula-se do erro na camada de saída

$$\delta^L = \nabla_a C \odot \sigma'(z^L)$$

// Retropropaga-se o erro por cada camada (backward pass)

para $l = L - 1$ **até** 2 **faça**

$$| \delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^L)$$

para $k = 1$ **até** K **faça**

$$| | \frac{\partial C}{\partial w_{jk}^l} = a^{l-1} (\sigma^l)^T$$

fim

$$| \frac{\partial C}{\partial b_j^l} = \sigma_j^l$$

fim

$$\nabla_{w,b} C = \langle \frac{\partial C}{\partial w_{jk}^l}, \frac{\partial C}{\partial b_j^l} \rangle \forall j, k, l$$

O algoritmo de backpropagation retomou a popularidade das redes neurais fazendo com que diversas novas propostas de modelos de redes neurais fossem desenvolvidas.

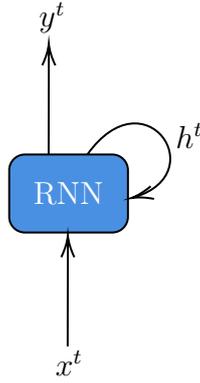


Figura 3.4: Rede Neural Recorrente

3.2 Redes Neurais Recorrentes

Redes Neurais Recorrentes [34] (Recurrent Neural Networks - RNNs) são um tipo de rede neural no qual estão presentes conexões recorrentes, ou seja, entre a saída de uma camada e sua entrada, ou de camadas anteriores. Este tipo de conexão permite que RNNs guardem representações internas dinâmicas de dependência entre estados. As Equações 3.26 e 3.27 representam uma RNN genérica, que pode ser visualizada através da Figura 3.4.

$$y^t = f(h^{t-1}, x^t) \quad (3.26)$$

$$h^t = g(h^{t-1}, x^t) \quad (3.27)$$

onde y^t é a saída da rede neural no instante t e h^t é o estado da rede neural no instante t . f e g representam a transformação da rede neural na entrada e no estado anterior para gerar, respectivamente, a próxima saída e o próximo estado.

Este tipo de rede neural é comumente utilizado para processar dados com correlação sequencial, em problemas como reconhecimento de fala [37] [38], classificação de texto [39] [40], representação de vídeos em texto [41] [42], modelagem acústica [43], entre outras. Em especial, para este trabalho, a aplicação de RNNs em problemas envolvendo séries temporais financeiras é particularmente relevante. Estes, vão desde indicação de tendência [44] [45] a previsão de preço ou retorno de ativos [46] [47] [48].

Existem diversos modelos de RNNs, que variam a maneira como as conexões recorrentes são feitas. A seguir são apresentados alguns dos modelos e aspectos relevantes ao estudo de RNNs.

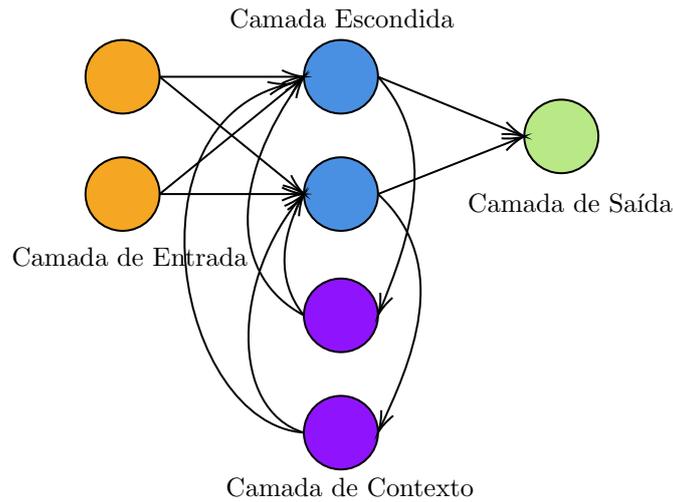


Figura 3.5: Rede Elman

3.2.1 Elman e Jordan

Dois dos primeiros e mais simples modelos de RNNs são o modelo Elman [49] e o modelo Jordan [50]. Ambos consistem em uma rede neural com uma camada de entrada, uma camada de saída, uma camada escondida e uma camada de contexto. Esta camada de contexto é a responsável por fazer as conexões recorrentes da rede neural. Na rede Elman a camada de contexto é conecta a saída da camada escondida na sua entrada. A rede Elman pode ser vista na Figura 3.5 e é representada pelas Equações 3.28 e 3.29.

$$y^t = \sigma_y(W_y h_{t-1} + b_y) \quad (3.28)$$

$$h^t = \sigma_h(W_h + U_h h_{t-1} + b_h) \quad (3.29)$$

A rede Jordan é bastante similar a Elman, sua diferença é que nela a camada de contexto, também conhecida como camada de estado, conecta a saída da rede a sua entrada. A rede Elman pode ser vista na Figura 3.6 e é representada pelas Equações 3.30 e 3.31.

$$y^t = \sigma_y(W_y h_{t-1} + b_y) \quad (3.30)$$

$$h^t = \sigma_h(W_h + U_h y_{t-1} + b_h) \quad (3.31)$$

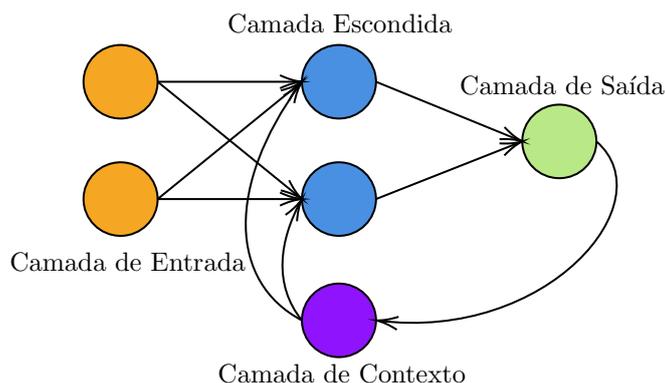


Figura 3.6: Rede Jordan

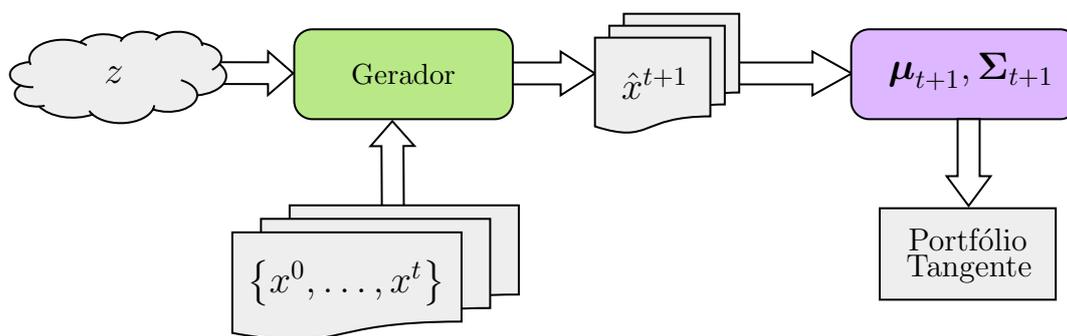


Figura 3.7: RNN desdobrada com $\tau = 3$

3.2.2 Treinamento de RNNs e o Problema de Longas Dependências

Para se treinar uma rede neural recorrente é necessário aplicar o que se chama de desdobramento da rede. Para desdobrar uma rede neural recorrente é necessário aplicar sua equação iterativamente por um número definido de passos τ . Desta forma a rede se transforma em uma rede acíclica de múltiplas camadas que compartilham os mesmos parâmetros. Para ilustrar é possível ver a rede neural da Figura 3.4 desdobrada com $\tau = 3$ na Figura 3.7. A rede desdobrada é descrita pela Equação 3.32, obtida através das Equações 3.26 e 3.27. Uma vez desdobrada a rede pode ser treinada normalmente, utilizando o algoritmo de backpropagation para calcular o gradiente e uma variante qualquer de gradiente descendente para minimizar a função custo.

$$y^3 = f(g(g(h^0, x^1), x^2), x^3) \tag{3.32}$$

onde h^0 é o estado inicial que deve ser determinado. Este é comumente iniciado com um vetor de zeros ou uma distribuição aleatória.

O problema no treinamento de uma rede neural desdobrada se encontra no fato

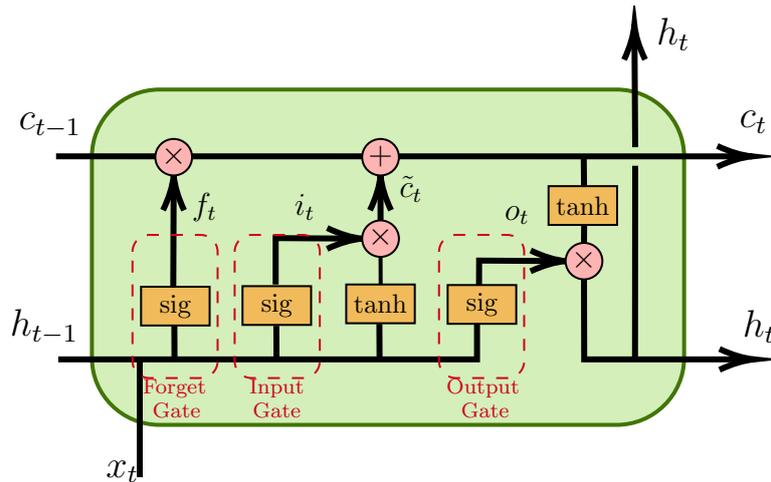


Figura 3.8: Célula LSTM

de que, quanto maior o numero de passos em que ela é desdobrada, mais instável se torna o gradiente. Este tende a convergir para zero (o mais comum) ou explodir. Isso faz com que os modelos mais simples de RNNs tenham dificuldade em detectar padrões em instantes do passado que sejam distantes. Este problema foi abordado por diversos autores na literatura [51] [52] [53].

Este comportamento do gradiente se deve ao fato de que os parâmetros da rede são compartilhados por todas as camadas da rede desdobradas. Com isso os pesos da rede são multiplicados por si próprios múltiplas vezes, Isto faz com que ou desapareçam ou explodam, dependendo da sua magnitude. Os modelos RNN de portas conseguem mitigar este efeito, apresentando memória muito maior do que modelos recorrentes simples.

3.2.3 Longa Memória de Curto Prazo

Atualmente a classe de modelos mais efetiva para a problemas sequenciais são as RNN de portas [54], na qual o modelo mais conhecido é o de Longa Memória de Curto Prazo (Long Short-Term Memory - LSTM) [55]. A solução apresentada por uma célula LSTM, para o problema do desaparecimento do gradiente, consiste em passar a informação do estado da LSTM de um passo para o seguinte sem que para isso ela precise ser multiplicada pelos pesos ou passar pela função de ativação da rede. Essa informação do estado é chamada de estado da célula (cell state). O fluxo do estado da célula entre diferentes passos de tempo é controlado pela porta de esquecimento (forget gate) e a porta de entrada (input gate). A Figura 3.8 ilustra a dinâmica de uma célula LSTM.

Como pode ser visto na Figura 3.8 todos os gates da célula LSTM consistem simplesmente em funções do tipo sigmoid, que vão retornar um valor entre 0 e 1.

Este valor vai variar com a combinação da entrada com o último estado escondido (hidden state). O cell state do passo anterior é multiplicado pela saída do forget gate, que tem a função de determinar o quanto do cell state anterior será mantido e o quanto será "esquecido". A seguir, o cell state é alterado pela entrada atual. O input gate controla o quanto a entrada atual irá alterar o cell state. Assim tem-se o cell state atualizado. Esse cell state é combinado e multiplicado pelos pesos e passa pela função de ativação para gerar a saída da célula LSTM. O output gate, então controla esse fluxo de saída. Desta forma obtém-se o hidden state atualizado, que também representa a saída da célula LSTM. Ele pode então ser direcionado para uma camada de saída da rede neural. Esse comportamento é descrito pelas Equações 3.33, 3.34, 3.35, 3.36, 3.37, 3.38.

$$f_t = \text{sigmoid}(w_f[h_{t-1}, x_t] + b_f) \quad (3.33)$$

$$i_t = \text{sigmoid}(w_i[h_{t-1}, x_t] + b_i) \quad (3.34)$$

$$o_t = \text{sigmoid}(w_o[h_{t-1}, x_t] + b_o) \quad (3.35)$$

$$\tilde{c}_t = \text{tanh}(w_c[h_{t-1}, x_t] + b_c) \quad (3.36)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (3.37)$$

$$h_t = o_t \odot \tanh c_t \quad (3.38)$$

As Equações 3.33, 3.34, 3.35, mostram o comportamento do forget gate, input gate e output gate, respectivamente. Como é possível ver, cada um dos gates possui seus próprios pesos e biases que serão otimizados via backpropagation e gradiente descendente. Desta forma, o modelo irá aprender a controlar o fluxo de informação no cell state de forma a garantir que informações mais relevantes possam ser "memorizadas" e as menos relevantes "esquecidas". A Equação 3.36 define \tilde{c}_t que é conhecido como candidato a cell state. Este juntamente com o cell state do passo anterior irão compor o cell state do passo seguinte ponderados pelo resultado do forget gate e input gate, respectivamente, como mostra a Equação 3.37. Por fim o próximo hidden state é calculado a partir do cell state ponderado pelo resultado do output gate, como mostra a Equação 3.38.

Como é possível ver na Equação 3.37, o cell state é calculado apenas por meio de uma transformação linear do cell state anterior, sem ser diretamente multiplicado por nenhum peso. Isso mostra que o erro propagado pelo algoritmo de backpropagation consegue fluir livremente sem sofrer o efeito de desaparecimento ou explosão causado por subsequentes multiplicações pelos parâmetros compartilhados, uma vez que a célula for desdobrada para o treinamento. É exatamente essa dinâmica que diferencia o modelo LSTM dos modelos simples de RNNs, tornando-o um dos

melhores candidatos para problemas sequenciais com dependências de longo prazo.

3.2.4 RNNs para Aplicações em Séries Temporais Financeiras

Como já foi previamente apresentado RNNs são amplamente utilizadas para problemas com dados que apresentam alguma correlação sequencial. O escopo deste trabalho visa conseguir representar o comportamento estatístico de séries temporais de portfólios de ativos do mercado financeiro. Para tal, buscou-se na literatura as aplicações de RNNs dentro deste escopo. Pode-se constatar a partir desta análise que são diversas as contribuições indicando que modelos de RNN são alternativas tão viáveis para este domínio quanto são para os demais domínios envolvendo dados sequenciais.

As aplicações de RNNs para séries temporais financeiras datam desde a década de 90. Kamijo et al. (1990) [56] desenvolveram um modelo para detectar padrões de triângulos em séries de ações da bolsa de Tóquio. Estes padrões são comumente utilizados em análise técnica como indicativos para tendências futuras de uma ação.

Saad et al. [45] Utilizaram 3 modelos de redes neurais, dentre eles uma RNN, para detecção de falsos alarmes referentes a sinais de queda no preço da ação. Desta forma pode-se aumentar a razão risco-recompensa ao reduzir tais falsos alarmes. O modelo RNN mostrou-se uma opção viável para o problema.

Rather et al. [47] desenvolveu um modelo híbrido composto por dois modelos lineares (média móvel auto-regressiva e suavização exponencial) combinadas com uma RNN para prever os retornos de ações. Os parâmetros do modelo foram otimizados por meio de algoritmo genético, obtendo resultados superiores aos dos benchmarks utilizados.

Chen et al. [57] desenvolveu um modelo utilizando LSTM para categorizar futuros retornos do mercado de ações da China em intervalos de retorno definidos. Quando comparado com uma categorização arbitrária, o modelo aumentou a acurácia de 14,3% para 27,2%, demonstrando a capacidade de LSTMs de extrair informações relevantes da série de retornos.

Huynh et al. [58] propôs um modelo usando uma Unidade Recorrente de Porta (Gated Recurrent Unit - GRU) bidirecional usando dados de notícias financeira combinados com dados históricos para previsão da mudança de preços do índice S&P 500 (composto pelas 500 maiores empresas do mercado dos Estados Unidos) e das ações que o compõe. O resultado foi comparado a dois dos modelos, até então estado da arte em sistemas de predição baseados em notícias [59] [60]. O modelo apresentou acurácia de 59,91%, enquanto ou outros modelos apresentaram acurácia de 55,21% [59] e 56,87% [60]. Mais uma vez o uso de RNNs, mais especificamente RNNs

de porta, apresentam a possibilidade de extrair informações das séries temporais financeiras que outros modelos não eram capazes.

Kim et al. [61] propõe um modelo híbrido para a previsão da volatilidade do principal índice de ações da bolsa da Coreia do Sul. Este modelo combina uma LSTM com modelos de Heteroscedasticidade Condicional Auto-regressiva Generalizada (Generalized Autoregressive Conditional Heteroskedasticity - GARCH), uma classe de modelos tradicionalmente utilizada para modelagem estatística de séries temporais. O modelo foi comparado com um modelo LSTM não combinado, variações de GARCH não combinados, redes neurais não recorrentes não combinadas e redes neurais não recorrentes combinadas com GARCH. Entre os modelos não combinados, a LSTM apresentou o menor MSE (47,01% menor que o segundo colocado). Dentre todos os modelos o combinados o modelo LSTM-GARCH foi o que apresentou menor MSE (57,3% menor que o segundo colocado). Estes resultados demonstram a capacidade de LSTM, tando de superar os resultados de redes neurais não recorrentes assim como o de modelos estatísticos já estabelecidos na área. Ainda vale destacar a possibilidade de potencialização dos resultados de LSTMs quando combinadas com outros modelos.

Zhou et al. [7] propôs um modelo para a previsão de preços de ações para operações de alta frequência (high-frequency trading), usando uma GAN composta por uma LSTM como gerador. O modelo foi comparado com diferentes modelos, como GARCH, redes neurais não recorrentes, LSTMs sem erro adversário, e Máquinas de Vetores de Suporte (Support Vector Machines - SVM). O modelo que combina GAN com LSTM apresentou Acurácia na Direção da Predição (Direction Prediction Accuracy - DPA) superior aos demais modelos em 71,16% das 378 simulações realizadas. O segundo colocado foi o modelo LSTM não combinado, apresentando maior DPA em 17,99% dos cenários. Esse resultado não só demonstra a aplicabilidade de LSTMs para modelar séries temporais de alta frequência, mas mais importante, mostra que modelos GANs podem potencializar o desempenho de LSTMs.

Estas referências mostram que modelos de RNNs, mais especificamente LSTMs ou outros modelos de portas, como GRUs, estão entre as melhores técnicas para se aplicar em problemas de séries temporais financeiras.

3.3 Redes Generativas Adversárias

Redes Generativas Adversárias (GANs) [6] são um framework para a estimação de um modelo generativo por meio de um processo adversário. GANs buscam modelar o comportamento de uma distribuição de probabilidade de forma implícita, sendo capazes de gerar amostras com o mesmo comportamento estatístico da distribuição

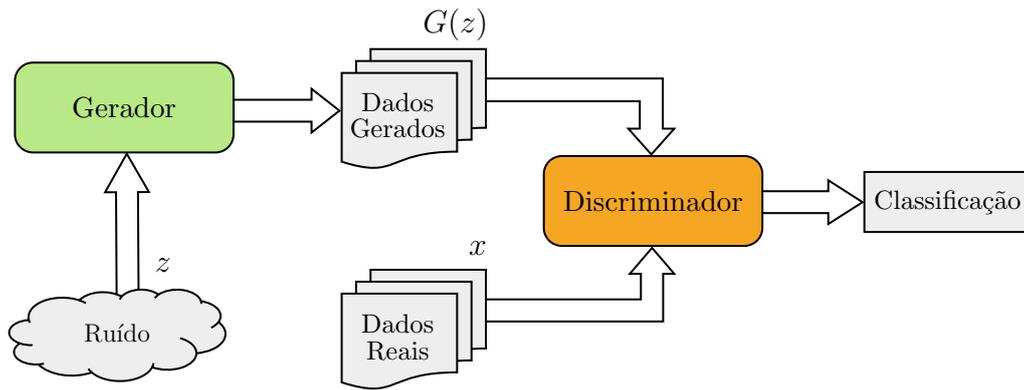


Figura 3.9: Rede Generativa Adversária

original. São amplamente aplicados em problemas de geração de amostras de distribuições complexas, como imagens.

Radford et al. [62] desenvolveram uma variante de GAN usando Redes Neurais Convolutivas (Convolutional Neural Network - CNN) de forma a conseguir gerar imagens realistas de um dado domínio quando treinado com outras imagens do mesmo domínio. Zhu et al. [63] desenvolveram uma aplicação para transformar desenhos simples em imagens fotorealistas. Wu et al. [64] desenvolveram um modelo para a geração de objetos 3D de diferentes categorias. Ledig et al. [65], aplicando GANs para aumentar a resolução de imagens, obteve resultados superiores ao de métodos tradicionalmente utilizados.

O processo adversário de treinamento envolve dois modelos: um modelo gerador denominado G e um modelo discriminador denominado D . O papel de G é o de gerar amostras que tentam aproximar o comportamento estatístico da variável que se deseja modelar. Já D avalia tanto amostras reais quanto amostras geradas, tentando classificá-las de acordo. O objetivo de G é maximizar o número de erros de classificação de D , enquanto o de D é classificar o maior número de amostras corretamente. O processo pode ser comparado com um cenário hipotético de um falsificador de quadros e um especialista em pinturas. O falsificador busca tornar suas pinturas cada vez mais próximas aos quadros originais, com a finalidade de poder enganar o especialista. Já o especialista busca melhorar sua capacidade de perceber características que o possibilitem distinguir entre um quadro verdadeiro e uma fraude. O processo de treinamento adversário é representado na Figura 3.9.

Dado um conjunto de amostras x provenientes da distribuição $p_r(x)$, tem-se como objetivo gerar amostras de p_G , tal que estas sejam estatisticamente indistinguíveis das amostras de $p_r(x)$. Para tal, define-se uma distribuição a priori de entradas de ruído $p_z(z)$, que é mapeada para o espaço dos dados por meio de $G(z; \theta_g)$, onde G é uma função com parâmetros θ_g . Define-se também $D(x; \theta_d)$ como uma função

com parâmetros θ_d . A saída $D(x)$ representa a probabilidade de x ter sido gerado pela distribuição p_r em vez de p_g . Treina-se D para maximizar a probabilidade de acerto ao categorizar uma amostra como proveniente de p_r ou p_g . G é treinado, simultaneamente, para minimizar $\log(1 - D(G(z)))$, ou seja, a taxa de acerto de D . Este processo pode ser representado como um jogo min-max de dois jogadores, como expresso na Equação 3.39.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(x)} [\log(1 - D(G(z)))] \quad (3.39)$$

O algoritmo 2 descreve o processo de treinamento de uma GAN onde o número k representa o número de vezes que o discriminador é treinado para cada vez que o gerador é treinado.

Algoritmo 2: Treinamento de uma GAN

```

para  $n = 1$  até total de iterações faça
  para  $k = 1$  até total de passos de treinamento do discriminador faça
     $z = m$  amostras de ruído de  $p_g(z)$ 
     $x = m$  amostras de dados de  $p_{\text{dados}}(x)$ 
    Atualizar o discriminador na direção ascendente do gradiente:
       $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x^i) + \log(1 - D(G(z^i)))]$ 
    fim
     $z = m$  amostras de ruído de  $p_g(z)$ 
    Atualizar o gerador na direção descendente do gradiente:
       $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^i)))$ 
  fim

```

O principal problema enfrentado pelas GANs é o problema da dificuldade de convergência durante o processo de treinamento. O treinamento de uma GAN exige encontrar o equilíbrio em um jogo de dois jogadores, diferentemente da maior parte dos algoritmos de otimização, que se baseiam em um processo contínuo de redução de uma função custo. Durante o treinamento de uma GAN, ao reduzir a função custo para um dos jogadores existe uma contrapartida de mover a função custo do outro no sentido contrário. Este processo iterativo pode levar a convergência, mas em muitos casos pode levar a resultados sem significado. Uma das formas mais comuns de não convergência no treinamento de GANs é o Colapso de Modo.

O colapso de modo consiste no caso em que o processo de treinamento da GAN leva o gerador a mapear diferentes pontos da entrada em um mesmo ponto na saída. Isso faz com que o gerador não consiga propriamente representar a distribuição de probabilidade desejada. O gerador passa a concentrar suas amostras em alguns poucos pontos que façam o discriminador errar. Uma vez que o discriminador consegue aprender que o gerador está tendo este comportamento, este apenas passa a deslocar

aqueles poucos pontos no espaço.

Diversas são as técnicas presentes na literatura para melhorar a convergência no processo de treinamento de GANs [66]. A seguir são apresentadas algumas das mais utilizadas e efetivas.

3.3.1 Wasserstein GAN

Visando mitigar a instabilidade no treinamento de GANs e problemas como o colapso de modo, várias técnicas para o treinamento de GANs foram concebidas. A mais conhecida entre elas é Wasserstein GANs (WGAN) [67]. Antes de analisar as diferenças entre o método tradicional de treinamento de GANs e o a WGAN faz-se necessário introduzir duas métricas que quantificam a similaridade entre duas distribuições de probabilidade.

A Divergência de Kullback-Leibler (KL) mede o quanto uma distribuição de probabilidade p diverge de uma segunda distribuição de probabilidade q . Ela é descrita pela Equação 3.40

$$D_{\text{KL}}(p \parallel q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx \quad (3.40)$$

Algo perceptível na Equação 3.40 é que a divergência de KL é assimétrica. Nos casos em que $p(x)$ é próxima de zero mas $q(x)$ é significativamente diferente de zero o efeito de $q(x)$ é praticamente desprezível. Isso pode causar problemas ao tentar avaliar similaridade de duas distribuições igualmente relevantes.

A Divergência de Jensen-Shannon é uma outra medida de similaridade, porém simétrica, diferentemente da divergência de KL. Ela é descrita pela Equação 3.41.

$$D_{\text{JS}}(p, q) = \frac{1}{2} D_{\text{KL}}(p \parallel \frac{p+q}{2}) + \frac{1}{2} D_{\text{KL}}(q \parallel \frac{p+q}{2}) \quad (3.41)$$

O método tradicional para o treinamento de GANs nada mais é do que a redução da Divergência de Jensen-Shannon entre a distribuição do gerador (p_g) e a distribuição real (p_r), conforme descrito por Goodfellow et al. (2014) [6]. Divergência de Jensen-Shannon, porém apresenta um problema para quantificar a similaridade entre distribuições que não possuem sobreposição. Arjovsky et al. (2017) [67] compara a divergência KL e a divergência de Jensen-Shannon a distância Wasserstein para o problema de medir a similaridade entre duas distribuições representando linhas paralelas. Ele mostra que a única a apresentar um valor de similaridade que varia de forma suave com a distância entre as linhas é a distância Wasserstein. Devido a forma suave com a qual a distância varia com a aproximação das distribuições ela apresenta bons resultados em relação ao aumento da estabilidade no processo de treinamento de uma GAN. A distância Wasserstein é descrita pela Equação 3.42.

$$W(p, q) = \inf_{\gamma \in \Pi(p, q)} \mathbb{E}_{(x, y) \sim \gamma} [\|x - y\|] \quad (3.42)$$

Ela é comumente ilustrada da seguinte forma. Supõe-se que a distribuição de probabilidade p é uma pilha de terra e deseja-se mover sua terra de forma que ela passe a ter o formato de q . Neste cenário a distância Wasserstein será o menor gasto de energia necessário para efetuar essa transformação. A função $\gamma(x, y)$ representa um plano de transporte, ou seja, a massa total que deve ser deslocada do ponto x para o ponto y da distribuição para cada possível ponto. É uma distribuição de probabilidade conjunta que tem como distribuições marginais p e q . O custo para fazer o deslocamento de x para y será então $\gamma(x, y) \|x - y\|$, ou seja, o total de massa a ser transportado multiplicado pela distância a ser percorrida. Desta forma, o custo total do plano de transporte γ será $\mathbb{E}_{(x, y) \sim \gamma} [\|x - y\|]$. A distância Wasserstein consiste então no menor custo de transporte dentro do conjunto de todos os possíveis planos de transporte, representados por $\Pi(p, q)$.

Como o calculo do ínfimo é extremamente custoso, pois exigiria estimar os custos de todos os possíveis cenários de transporte o calculo da distância Wasserstein é feito por meio da dualidade Kantorovich-Rubinstein [68] conforme mostra Equação 3.43.

$$W(p, q) = \sup_{\|f\|_L \geq K} \mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{x \sim q}[f(x)] \quad (3.43)$$

de forma que o supremo é calculado no conjunto de todas as funções f que satisfaçam a condição de K-Lipschit, ou seja $\|f\|_L \geq K$. Agora suponto que a função f venha a partir de uma família de funções K-Lipschit contínuas, $\{f_w\}_{w \in W}$, parametrizadas em w . Desta forma a função custo a ser minimizada para a WGAN é dada pela Equação 3.44

$$L(p_r, p_g) = W(p_r, p_g) = \max_{w \in W} \mathbb{E}_{x \sim p_r}[f_w(x)] - \mathbb{E}_{x \sim p_g}[f_w(G(z))] \quad (3.44)$$

O principal problema é garantir a continuidade de Lipschit em f_w . Para tal, restringe-se o valor dos pesos w a um intervalo definido $[-c, c]$, após cada atualização do gradiente. Desta forma é garantido que f_w possua um limite superior e inferior, de forma a preservar a continuidade de Lipschit. Está é uma maneira simples de garantir a continuidade de Lipschit, porém apresenta o problema de que o modelo torná-se muito sensível ao hiperparâmetro c . Isso faz com que, em muitos casos, o modelo ainda apresente dificuldade de convergir. Além disso, essa restrição dos pesos reduz a capacidade da GAN de modelar funções complexas.

O modelo chamado de Wasserstein GAN com Penalidade de Gradiente (Wasserstein GAN with Gradient Penalty - WGAN-GP) [69] faz uso de uma penalidade de gradiente como forma de manter a continuidade de Lipschit e mitigar os problemas

causados pelas restrição dos pesos.

Uma função f é 1-Lipschit se e somente se ela possuir gradientes com norma menor ou igual a 1. Desta forma, no lugar de restringir os pesos do modelo a WGAN-GP penaliza o modelo caso seu gradiente ultrapasse o limite de 1. A Equação 3.45 representa a função custo de uma WGAN-GP, que é igual a de uma WGAN com o acréscimo da penalidade por gradiente. A função f_w já é conotada como o discriminador e representada por D .

$$L(p_r, p_g) = \mathbb{E}_{x \sim p_g}[D(G(z))] - \mathbb{E}_{x \sim p_r}[D(x)] + \lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (3.45)$$

onde \hat{x} é amostrada de $G(z)$ e x com $0 \leq t \leq 1$ como descrito na Equação 3.46.

$$\hat{x} = tG(z) + (1 - t)x \quad (3.46)$$

Tanto WGANs quanto WGAN-GPs são alternativas ao método original de treinamento de GANs que visam melhorar a convergência do modelo, mantendo o treinamento mais estável. Porém, uma comparação entre a qualidade de imagens geradas por WGANs, WGAN-GPs e uma GAN Convolutiva Profunda (Deep Convolutional GAN - DCGAN) [62], um dos principais modelos para a geração de imagens que utiliza o o processo padrão de treinamento, mostra que o DCGAN conseguiu gerar imagens com qualidade superior as dos modelos WGANs e WGAN-GPs [66]. A DCGAN utiliza outras técnicas para auxiliar na convergência, mas este resultado mostra que a escolha pelo melhor método para treinar uma GAN irá variar dependendo da aplicação.

3.3.2 Discriminação de Mini-lotes

Um dos principais problemas no treinamento de GANs, como já foi descrito, é o colapso de modo. Como as amostras do gerador são avaliadas independentemente pelo discriminador torna-se mais difícil escapar do colapso de modo e o gerador tende a ser incentivado a gerar amostras de um único ponto que pareçam realistas para o discriminador. Uma vez que o discriminador aprende que aquele ponto está sendo gerado seu gradiente passa a deslocar este ponto no espaço indefinidamente, de forma que o treinamento não converge. Uma estratégia para evitar este problema é combinar múltiplas amostras do gerador e apresenta-las ao discriminador. Este procedimento é chamado de discriminação de mini-lotes (Minibatch discrimination) [66].

Para modelar a proximidade entre amostras de um mini-lote defini-se $f(x_i) \in \mathbb{R}^A$ como um vetor de variáveis para a entrada x_i , produzidas por uma camada

intermediária do discriminador. Este vetor é então multiplicado por um tensor $T \in \mathbb{R}^{A \times B \times C}$, resultando na matriz $M_i \in \mathbb{R}^{B \times C}$. A Equação 3.47 descreve a similaridade entre as amostras do minibatch.

$$c_b(x_i, x_j) = \exp(-\|M_{i,b} - M_{j,b}\|_{L_1}) \quad (3.47)$$

A saída $o(x_i)$ da camada de mini-lote para a amostra x_1 é definida pela soma $c_b(x_i, x_j)$ para as demais amostras nas Equações 3.48 e 3.49.

$$o(x_i)_b = \sum_{j=1}^n c_b(x_i, x_j) \quad (3.48)$$

$$o(x_i) = [o(x_i)_1, o(x_i)_2, \dots, o(x_i)_B] \in \mathbb{R}^B \quad (3.49)$$

Essa saída é então concatenada as demais variáveis da camada intermediária e usadas como entrada para a próxima camada.

A discriminação por mini-lote apresentou resultados de imagens visualmente realistas e com resultados expressivos quando utilizadas para o treinamento semi-supervisionado, superando inclusive modelos consolidados como o DCGAN. Isto demonstra o potencial da técnica de discriminação por mini-lote em melhorar o treinamento de modelos GAN.

3.3.3 Outras Técnicas para o Treinamento de GANs

Diversas outras técnicas foram propostas na literatura com a finalidade de melhorar a estabilidade do treinamento de GANs, assim como de evitar efeitos como o colapso de modo.

O Casamento de Variáveis (Feature Matching) [66] propõe uma alteração no objetivo do gerador. No lugar de buscar aumentar o erro de classificação do discriminador, o gerador é treinado para gerar dados que façam com que as variáveis de uma camada intermediária do gerador apresentem comportamento equivalente ao que apresentariam com dados reais. Aplicações empíricas mostraram a técnica como efetiva em casos que o treinamento da GAN se torna instável. O autor destaca, porém, que no quesito de geração de imagens realistas a aplicação de discriminação por mini-lote é superior ao casamento de variáveis.

A Normalização por Mini-lotes (Batch Normalization) [70] consiste na normalização da entrada de cada camada escondida e já era aplicada a RNNs com a finalidade de acelerar o tempo de treinamento. Sua aplicação, porém, também se mostrou relevante para o treinamento de GANs, tendo sido popularizada por ser utilizada pela DCGAN [62], um dos principais benchmarks de modelo generativo para imagens. Apesar da popularidade, segundo Salimans et al. [66] a normalização

por mini-lotes é menos efetiva que a discriminação por mini-lotes para a evitar o colapso de modo.

A Suavização de Rótulo é uma técnica comumente utilizada em redes neurais, inclusive para reduzir a vulnerabilidade a exemplos adversários [71]. Ela consiste em aplicar valores suavizados as classes de treinamento, no lugar de 0 e 1. Essa aplicação também é válida para GANs, com a diferença que no caso de GANs a suavização deve ser feita apenas em rótulos positivos. Os negativos devem ser mantidos como 0, pois tê-los diferente de 0 afeta negativamente o discriminador [70].

3.3.4 GANs Aplicadas a Séries Temporais

GANs tem como seu principal domínio de aplicação geração de imagens. As aplicações para geração de imagens são extensas e já foram apresentadas neste trabalho. O escopo deste projeto, porém, está na proposta de um modelo que seja capaz de reproduzir o comportamento estatístico de séries temporais financeiras, ou seja, é necessário que o modelo seja capaz de considerar todo o histórico de retornos para conseguir reproduzir de forma realística o comportamento estatístico futuro. GANs também apresentam um resultado expressivo na geração de sequências, incluindo séries temporais de ações.

Lotter et al. [72] utilizou uma GAN composta por LSTM para gerar o próximo frame de uma sequência de vídeos sintética. Foi feito uso tanto do erro adversário, quanto do MSE em relação aos dados reais, de forma a agilizar o treinamento. Pode-se constatar que o uso do erro adversário fez as imagens conterem mais detalhes do que o modelo treinado apenas com MSE.

Yu et al. [73] desenvolveu um modelo de GAN composto por uma rede neural convolucional (Convolutional Neural Networks - CNN) para geração de sequências, tanto de texto, quanto de música. O modelo de texto foi treinado tanto na geração de poemas Chineses quanto na de discursos políticos. O modelo para geração de música foi treinado com um dataset de 695 músicas folk. Os resultados obtidos foram superiores aos obtidos com uma LSTM treinada com máxima verossimilhança (Maximum likelihood estimation - MLE).

Lamb et al. [74] propôs uso de GANs para auxiliar o treinamento de redes neurais recorrentes de forma a forçar que a dinâmica da rede se mantenha a mesma operando com dados reais ou dados retroalimentados.

Esteban et al. [75] utilizou uma GAN na geração de dados médicos de UTIs, mais especificamente saturação de oxigênio, taxa de batimento cardíaco, taxa respiratória e pressão arterial média. Como forma de avaliar se o modelo realmente convergiu para a distribuição de probabilidade foi realizado o treinamento de um outro modelo utilizando os dados gerados durante o treinamento e os dados reais para teste.

Zhou et al. [7], como já apresentado na Subseção 3.2.4 propôs um modelo para a previsão de preços de ações para high-frequency trading, usando uma GAN composta por uma CNN como discriminador e uma LSTM como gerador. Ele usa como função de custo não apenas o erro adversário, mas também, o erro absoluto da previsão e uma função de custo relacionada a direção da previsão. O modelo apresentou resultados nas simulações 74,72% superiores a uma LSTM sendo treinada sem o erro de adversário. Tais resultados mostram que GANs podem ser usadas como um bons preditores para o preço de ativos financeiros. Cabe avaliar se as amostras geradas pelo modelo também conseguem ser usadas a fim de estimar a média e a covariância amostral do próximo instante de tempo de uma série de retornos, o que é o objeto de estudo deste trabalho.

Chen et al. [76] utilizou um modelo GAN, composto por CNNs, para a geração de cenários para plantas de energias renováveis. O modelo se mostrou capaz de gerar perfis realísticos para energia fotovoltaica e eólica. O modelo também foi capaz de gerar correlações espaço-temporais ao ser treinado com dados de diferentes localizações geográficas.

Gupta et al. [77] propôs a aplicação de um modelo GAN para prever a trajetória de pedestres caminhando. Esta aplicação é de vital importância para plataformas autônomas móveis, como carros autônomos ou robôs. O modelo é composto por LSTMs tanto no gerador quanto no discriminador, além de um módulo composto por uma MLP para codificar e decodificar as trajetórias. O modelo foi treinado com uma combinação do erro adversário com uma função de custo do tipo L2.

Toda a literatura apresentada acima aponta o uso de GANs como um boa candidato para modelar o processo estocástico dos retornos das ações para o problema de seleção de portfólios.

Capítulo 4

Arquitetura do Modelo e Viabilidade

Como foi apresentado no Capítulo 3 investidor diversifica seu investimento em diferentes ativos de forma a reduzir sua exposição ao risco. A seleção de um portfólio de ativos é um problema que exige do investidor conseguir modelar, não somente o valor esperado do retorno dos ativos presentes no universo de ativos do portfólio, mas também inferir sua variância e a correlação entre estes ativos.

Diversos são as propostas de modelos preditivos aplicados ao mercado financeiro. A Subseção 3.2.4 mostra que modelos de RNNs, mais especificamente LSTMs e outros modelos RNNs de portas como GRUs, apresentam excelentes resultados como modelos preditivos. Estes modelos, porém, cobrem apenas a parte da estimação do valor esperado num problema de seleção de portfólios. Faz-se ainda necessária a estimação da variância e da correlação entre os ativos. O Capítulo 5 apresenta diversas técnicas utilizadas para obter melhores estimações para a matriz de covariância dos ativos em um portfólio, porém fazem fortes simplificações a respeito do processo estocástico das séries temporais financeiras. A matriz de covariância é então comumente calculada com utilizando-se as amostras do retorno passado. A proposta deste trabalho é apresentar um modelo que seja capaz de utilizar as amostras do passado com o único fim de estimar a densidade de probabilidade do próximo instante de tempo de forma implícita e então usá-las para obter uma estimação mais confiável da matriz de covariância para este instante de tempo futuro.

GANs, como foi apresentado no no Capítulo 3, são o estado da arte em modelos geradores, conseguindo modelar de forma implícita o comportamento estatístico de variáveis aleatórias complexas de forma a ser capaz de gerar imagens com as mesmas características estatísticas das amostras originais. Este trabalho pretende utilizar um modelo GAN para gerar amostras de ativos em um dado universo de ativos e, por meio destas amostras, determinar os portfólios eficientes para o dado universo. O modelo é composto por RNNs, mais especificamente LSTM, de forma a conseguir

fazer uso da capacidade destas redes de guardar longas dependências temporais, uma vez que estas já apresentam bons resultados em problemas preditivos de séries temporais financeiras.

Este capítulo aborda o processo para a consolidação da arquitetura deste modelo, discutindo diversas propostas de arquitetura e processos de treinamento avaliados.

4.1 RNNs para a Geração de Séries Temporais

RNNs se apresentam como o estado da arte para problemas referentes a dados com correlação temporal, possuindo diversas aplicação para a predição de séries temporais, incluindo séries temporais financeiras. RNNs também podem ser utilizadas para a geração de sequências, uma vez que pode-se utilizar a saída de uma RNN como entrada no próximo instante de tempo e repetir este processo. Para conseguir gerar amostras diferentes, dado um mesmo conjunto de dados de instantes anteriores, basta utilizar uma variável aleatória como estado inicial. O problema com estas amostras é que elas não vão refletir o comportamento estatístico do processo estocástico em questão. Isso se deve ao fato de que as RNNs apenas geram sequências nas quais o próximo instante é o mais provável dados os instantes passados. Isso ocorre uma vez que as RNNs são treinadas para reduzir uma função custo como o MSE, por exemplo, de forma que elas acabam se tornando boas estimadoras do valor esperado de uma série temporal no seu próximo instante.

Por mais que a inviabilidade da aplicação de um modelo RNN para a geração de amostras para o próximo instante de tempo de um processo estocástico seja algo possível de se avaliar a priori, conforme foi discutido no parágrafo anterior, foi montado um experimento com dados sintéticos para confirmar esta hipótese. Foi treinada uma rede neural composta por uma camada LSTM conectada a uma camada completamente conexa com um neurônio de saída.

A série temporal utilizada foi um processo estocástico gaussiano estacionário, ou seja, cada instante de tempo consiste em uma amostra de uma variável gaussiana, conforme mostra a Figura 4.1 e a Figura 4.2. A RNN foi treinada por 200 épocas para minimizar o MSE usando Estimativa de Momento Adaptativo (Adaptive Moment Estimation - Adam) [78], uma variante de gradiente descendente muito popularizada para o treinamento de redes neurais por apresentar melhores resultados superiores ao de outras variações [78].

Como pode-se observar na Figura 4.3 o treinamento da LSTM convergiu para um erro médio próximo de 0.01. Tendo em vista que a distribuição analisada possui variância de 0.01 este é o menor erro médio possível. Conforme o esperado, a RNN treinada gera amostras próximas ao valor esperado da gaussiana utilizada. Ao retroalimentando a saída da rede na sua entrada é possível observar que a série

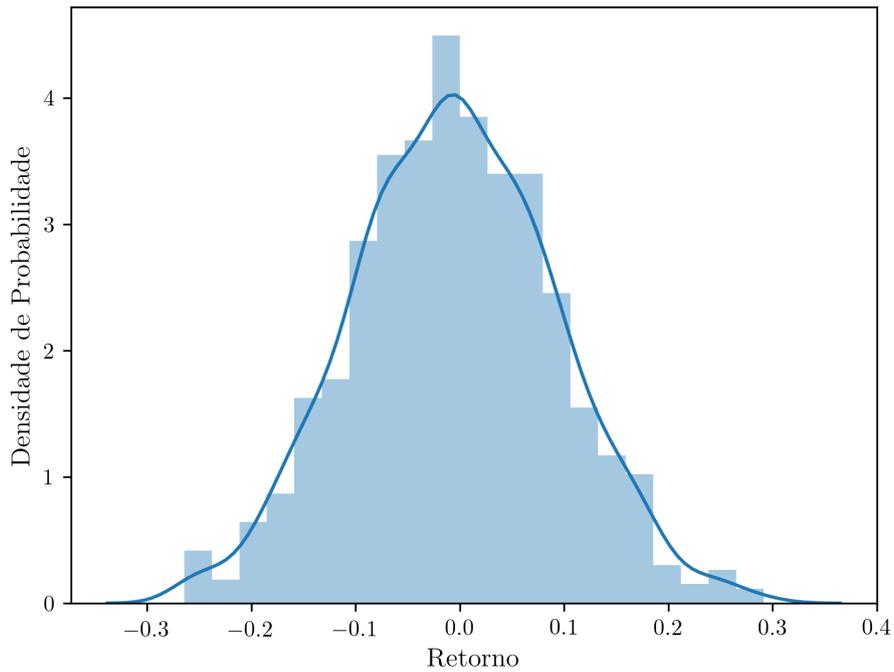


Figura 4.1: Distribuição do Processo Estocástico Gaussiano Estacionário no instante t

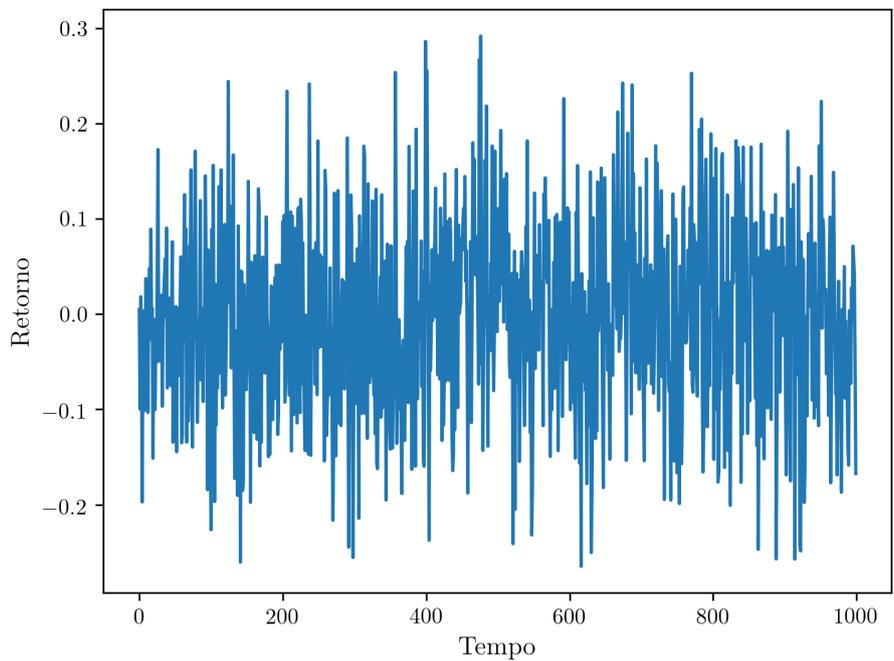


Figura 4.2: Ocorrência do Processo Estocástico Gaussiano Estacionário

gerada apresenta um comportamento muito distante da série original, como pode ser visto na Figura 4.4. A densidade de probabilidade das amostra geradas pode ser visto na Figura 4.5. Pode-se observar que a densidade de probabilidade para

um dado instantâneo da série é uma gaussiana com variância muito diferente da obtida pelos dados gerados pela rede.

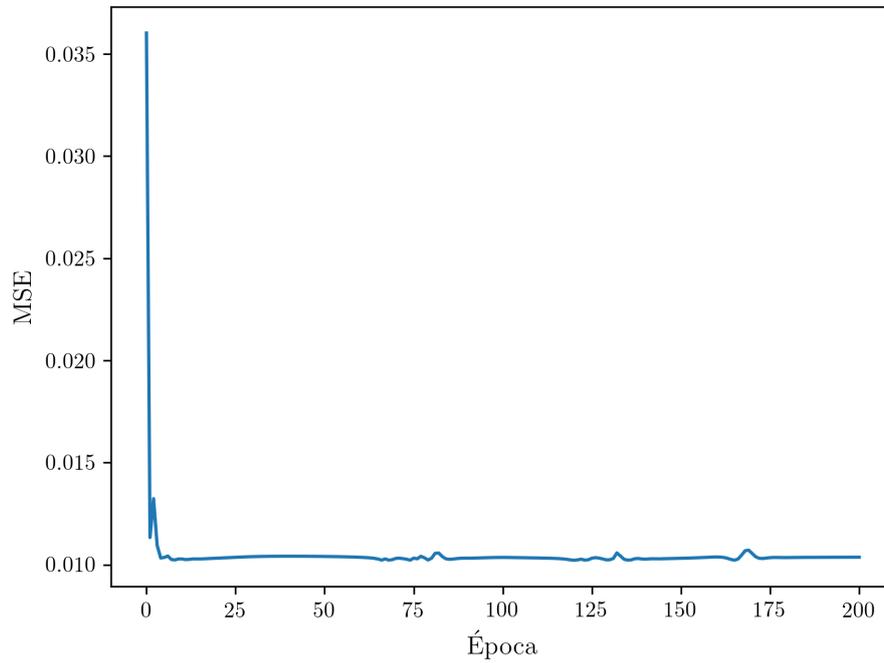


Figura 4.3: Evolução do treinamento da LSTM. Pode-se observar que o treinamento convergiu por um erro médio próximo ao mínimo global.

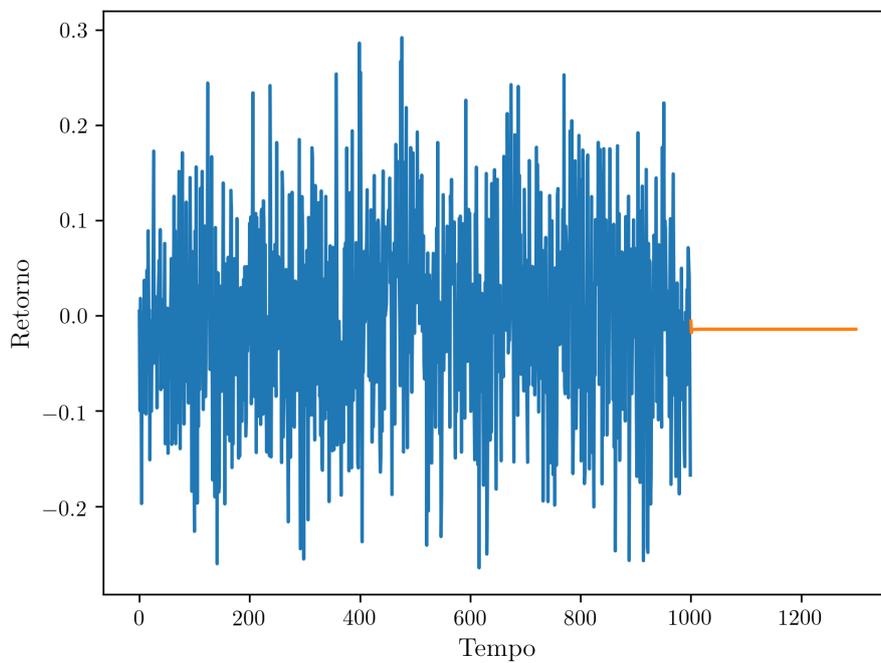


Figura 4.4: Série temporal gerada por uma LSTM

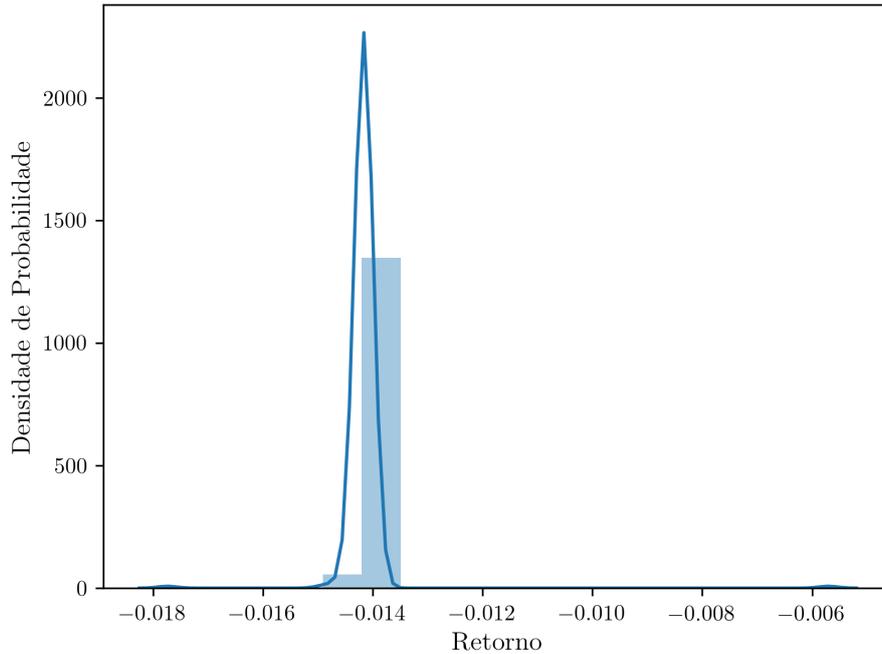


Figura 4.5: Distribuição das amostras geradas por uma LSTM

Com base nesse experimento confirma-se a hipótese de que, apesar de LSTMs apresentarem ótimos resultados para estimar o valor esperado para o próximo instante de tempo de uma série temporal, faz-se necessário outra forma de estimar a variância da mesma.

4.2 GANs para a Geração de Séries Temporais

Para avaliar a viabilidade da aplicação de GANs para a geração de amostras provenientes de processos estocásticos correlacionados foram conduzidos experimentos com diferentes arquiteturas, usando como base dados sintéticos.

4.2.1 Dados Sintéticos

De forma a mais facilmente avaliar a convergência dos modelos propostos optou-se por inicialmente fazer uso de dados sintéticos, de forma a ser possível saber com exatidão qual a distribuição de probabilidade esperada para cada instante de tempo.

Segundo a Hipótese do Mercado Eficiente (HME) [79], os preços dos ativos refletem completamente a informação disponível. A conclusão desta premissa é de que seria impossível para investidores comprarem ativos subvalorizados ou venderem ativos sobrevalorizados. Existem diversos argumentos contra a HME, como os próprios trabalhos sobre modelos preditivos aplicados a séries temporais financeiras, porém a HME apresenta modelos explícitos para o comportamento de ativos financeiros que

servem como um bom ponto de partida para avaliar a viabilidade de um modelo. Uma das formas comumente utilizadas para modelar o comportamento dos ativos, segundo a HME, afirma que os preços dos ativos seguem um passeio aleatório [79], sendo inclusive utilizada em um dos modelos mais populares para a precificação de opções, o modelo de Black-Scholes [80]. Ou seja, o preço do próximo instante de um ativo nada mais é do que seu preço anterior, acrescido de uma variável aleatória gaussiana.

Para o problema de seleção de portfólio a série que leva-se em consideração é a dos retornos do ativo. Dado que o preço do ativo foi modelado como um passeio aleatório, seu retorno será consequentemente um processo estocástico gaussiano. Como o trabalho trata de um portfólio é necessário modelar também a relação entre os ativos. Para isso fixou-se uma matriz de correlação. Tendo então estes processos estocásticos estacionários correlacionados entre si, pode-se aplica-los para avaliar o comportamento dos modelos propostos. A Figura 4.6 mostra duas séries temporais, referentes a uma realização dos processos estocásticos dos dois ativos presentes no portfólio. A Figura 4.7 mostra a distribuição de probabilidade conjunta em um dado instante de tempo.

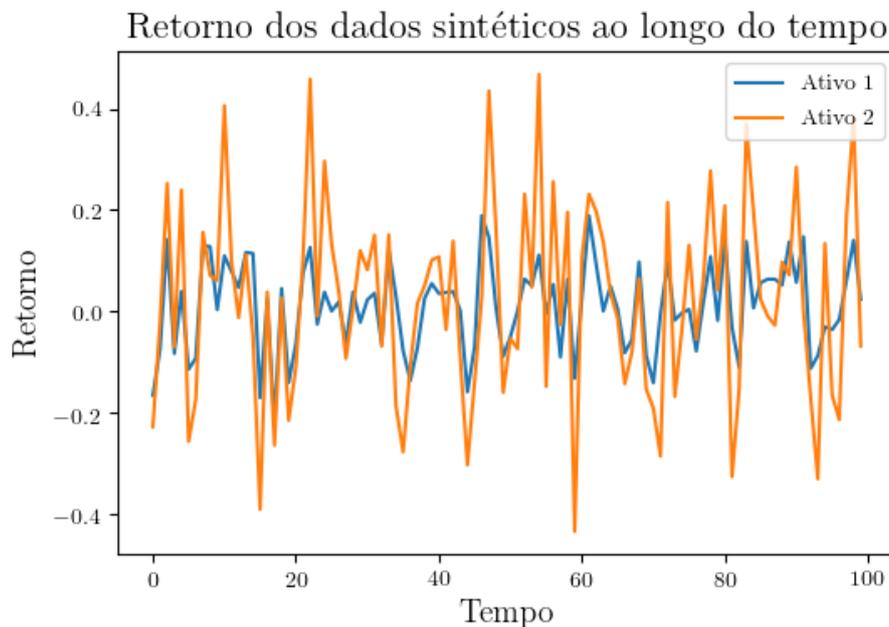


Figura 4.6: Séries temporais dos ativos do um portfólio sintético

4.2.2 GAN Composta por LSTM

Tomando como base modelos previamente abordados no Capítulo 3 que aplicam GANs a dados com correlações temporais [72] [75] [7] [77] optou-se por um modelo de

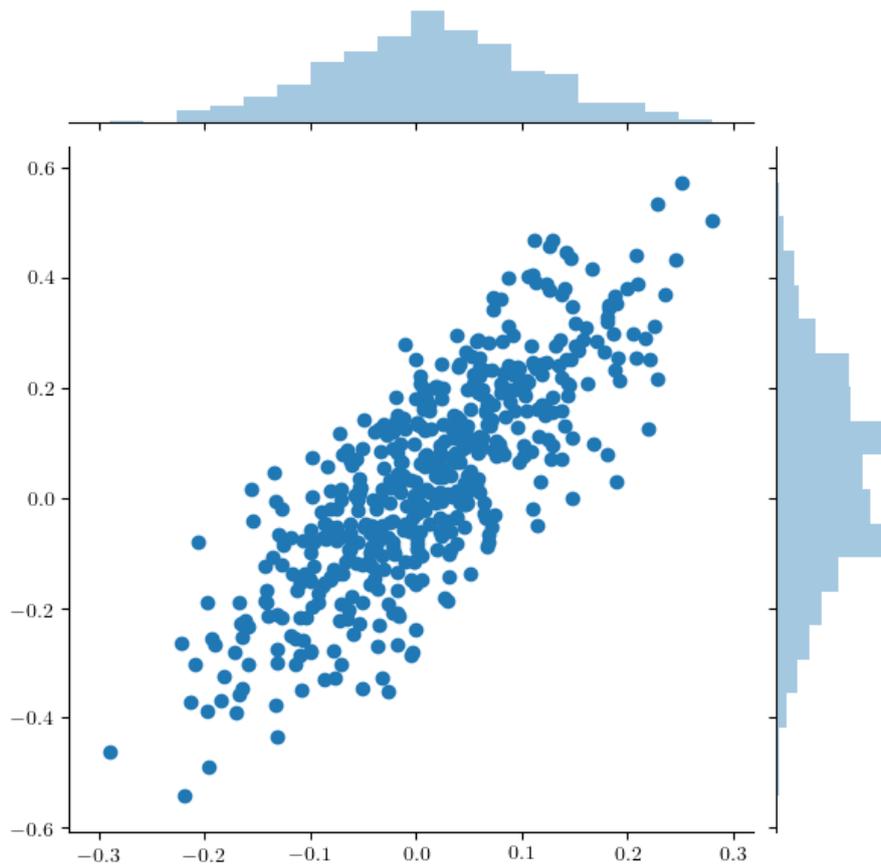


Figura 4.7: Distribuição de probabilidade conjunta dos ativos de um portfólio num instante de tempo t

GAN usando uma LSTM como gerador. O gerador utilizado consiste em uma RNN contendo uma camada LSTM ligada a uma camada de saída totalmente conectada. Este gerador foi treinado de forma a combinar o erro adversário do gerador com o MSE calculado com base nas amostras geradas e nos dados sintéticos utilizados. O uso do MSE ou alguma outra métrica de erro que leva em consideração a diferença entre os dados reais e os gerados é algo utilizado em aplicação de GAN para séries temporais como forma de auxiliar a convergência do modelo [72] [75] [7] [77].

Para o discriminador foram testadas duas arquiteturas diferentes: um discriminador com a mesma topologia do gerador e uma CNN. Isto se deve ao fato de que, como o discriminador só é utilizado durante o treinamento, só é necessário que ele seja capaz de manter dependências temporais do mesmo tamanho da janela de amostra do treinamento, não tornando tão fundamental o uso de um modelo RNN. LSTMs, por mais que sejam treinadas com janelas de tamanho fixo, conseguem generalizar o treinamento de forma a guardar dependências por períodos maiores. Para exemplificar, uma LSTM que consegue extrair a informação de que caso um parêntese seja aberto em um texto o mesmo deve ser fechado durante um treina-

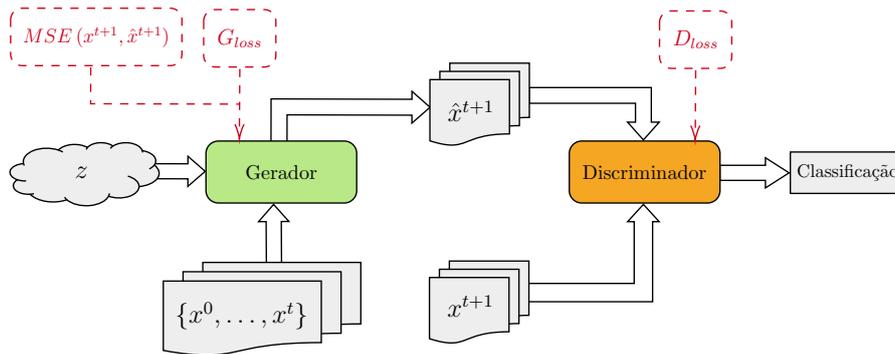


Figura 4.8: Modelo GAN para séries temporais

mento com sequências de tamanho 100, vai ser capaz de fechar parênteses abertos a distâncias maiores do que 100 durante sua execução. Como apenas o gerador vai ser usado em execução, essa capacidade de manter longas dependências não é fundamental para o discriminador.

O erro adversário utilizado foi o erro original da GAN [6], onde o discriminador vai buscar minimizar o erro de classificar se uma amostra é real ou gerada e o gerador vai buscar aumentar o erro do discriminador. O erro do gerador combinou o erro adversário com o erro MSE por meio de uma média geométrica, uma vez que os erros possuem grandezas distintas.

O modelo, que pode ser visto na Figura 4.8 foi treinado durante 1000 épocas com um conjunto de dados sintéticos de 5000 amostras, das quais 4000 foram utilizadas para treinamento e 1000 para teste. Este número de amostras foi escolhido por se aproximar do tamanho da amostra dos dados reais disponível. Os dados foram agrupados em janelas para o treinamento da LSTM, sendo testadas janelas de períodos comumente utilizados na análise técnica de ativos financeiros (20, 50, 100). Essas janelas foram agrupadas em lotes de 10 janelas, de forma que em cada época de treinamento o gradiente é calculado e os pesos são alterados para cada lote. Foi utilizado Adam gradiente descendente durante o treinamento com taxa de aprendizado, sendo avaliadas um intervalo com taxas de aprendizado comumente utilizadas: 0,0001, 0,0005, 0,001, 0,005, 0,01.

Avaliou-se tanto o comportamento do gerador para gerar amostras do próximo instante de tempo quanto a de continuar a série temporal uma vez que a saída é realimentada. O resultado obtido foi inesperado, uma vez que o modelo convergiu para gerar amostras muito semelhantes para o próximo instante de tempo, porém foi capaz de uma vez realimentada a saída, a série gerada apresentou o comportamento esperado. Como os dados sintéticos que foram utilizados foram gerados por uma série estacionária, esperava-se obter uma distribuição igual tanto no tempo quanto no espaço de probabilidade. Como pode ser visto na Figura 4.10 as amostras geradas

para um mesmo instante de tempo encontram-se concentradas já na Figura 4.9 observa-se que os pontos estão distribuídos acompanhando a distribuição esperada.

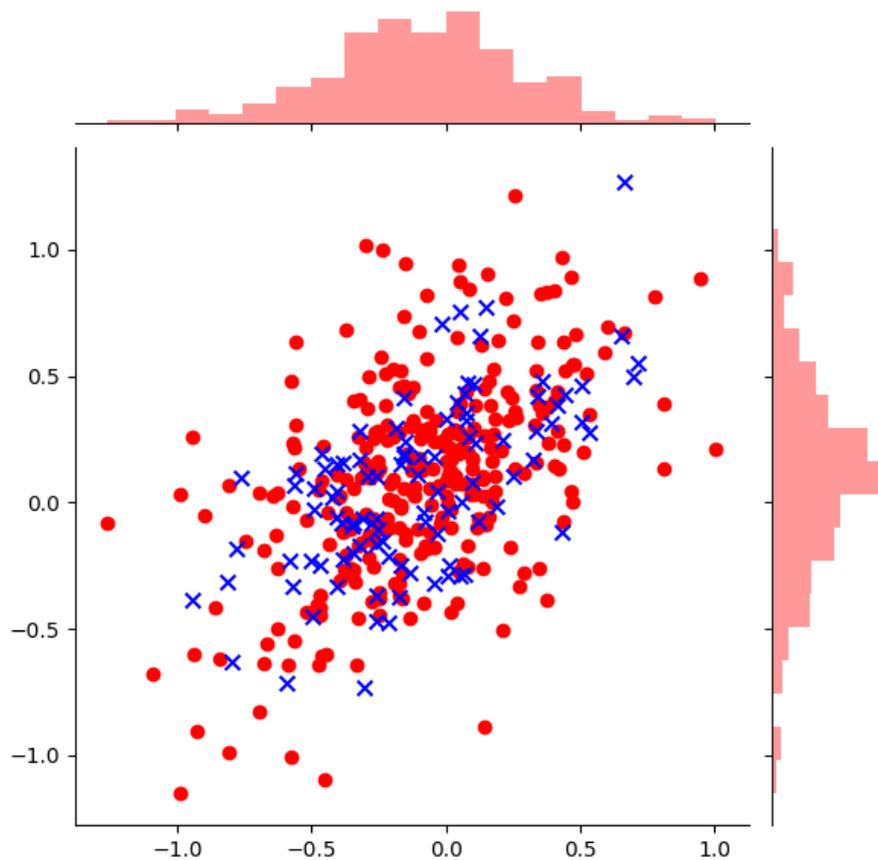


Figura 4.9: Distribuição dos dados gerados para os próximos 100 instantes (em vermelho) em contraste com a distribuição esperada para os dados sintéticos (em azul)

Quando comparado com uma LSTM simples na geração de séries temporais o modelo GAN ao ser conseguiu gerar amostras sucessivas consistentes com a estatística da série temporal. Essa capacidade de utilizar GANs para auxiliar no treinamento de uma LSTM que irá operar em regime de retroalimentação já foi demonstrado por Lamb et al. [74]. Porém para geração de amostras do mesmo instante de tempo a GAN não conseguiu gerar variabilidade entre as amostras. A falta de diversidade entre as amostras ao aplicar GANs para séries temporais também foi relatado no trabalho de Gulpta et al. [77]. A GAN comportou-se como se o vetor aleatório z , que é usado como entrada do gerador estivesse sendo desconsiderado. Lotter et al. [72] já havia verificado seu modelo para a previsão do próximo frame de um vídeo não apresentou diferença ao utilizar ou não o vetor z , porém como o modelo buscava apenas estimar o próximo frame de forma determinística o uso de GAN serviu ao propósito de que a imagem gerada tivesse maior qualidade. Outra aplicação de

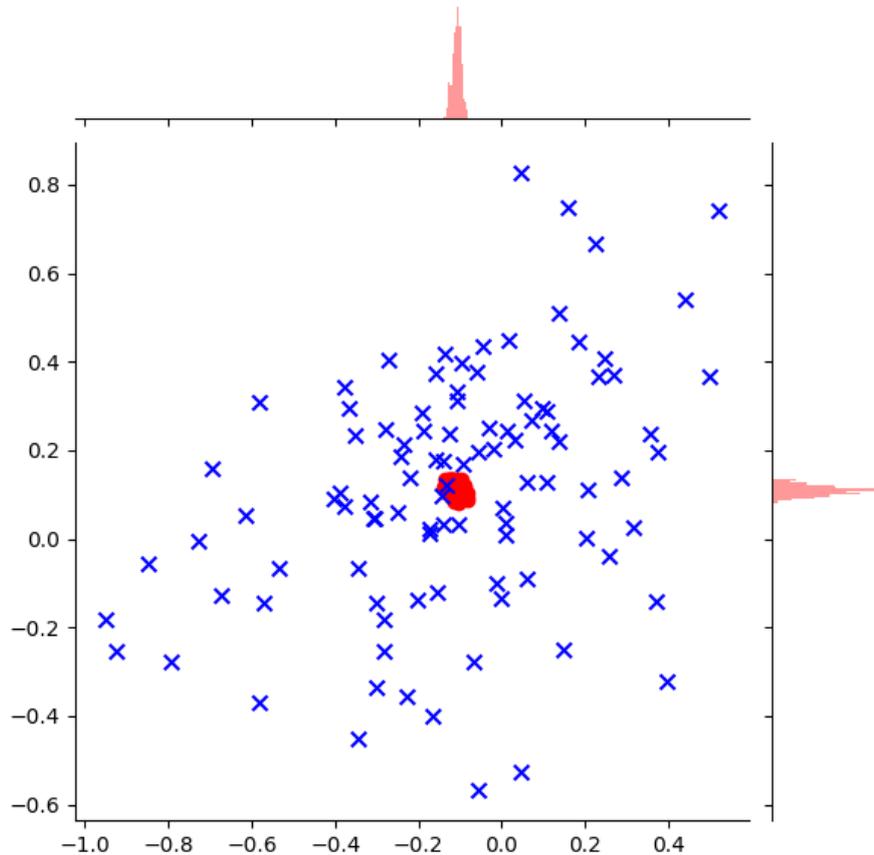


Figura 4.10: Distribuição dos dados gerados no instante ($t = 1$) (em vermelho) em contraste com a distribuição esperada para os dados sintéticos (em azul)

GANs para séries temporais, mas que tinha o intuito de usar a GAN como modelo preditivo foi a de Zhou et al. [7], usando GAN para estimar o preço de ativos do mercado financeiro, de forma que a variabilidade das amostras em um instante de tempo não precisou ser avaliada. Por último, o modelo de Esteban et al. [75] consistiu em gerar amostras de séries temporais médicas, retroalimentando a saída da GAN. Conforme foi observado que a GAN foi capaz de gerar séries que no tempo apresentam o mesmo comportamento da série original. Desta forma Esteban et al. também não precisaram avaliar a variabilidade estatística dentro de cada instante de tempo.

A própria natureza do problema pode ser a causa deste comportamento. Modelos CGAN [81] utilizam uma variável categórica para condicionar a distribuição da GAN, desta forma o modelo possui diversas amostras por categoria de forma a conseguir generalizar o comportamento. No caso de séries temporais deseja-se estimar a probabilidade condicional a uma determinada janela do passado. O problema é que, para isso, só tem-se a disposição a única ocorrência na realidade para aquele dado. Tendo isso em vista, a componente de MSE no erro da GAN propicia que o

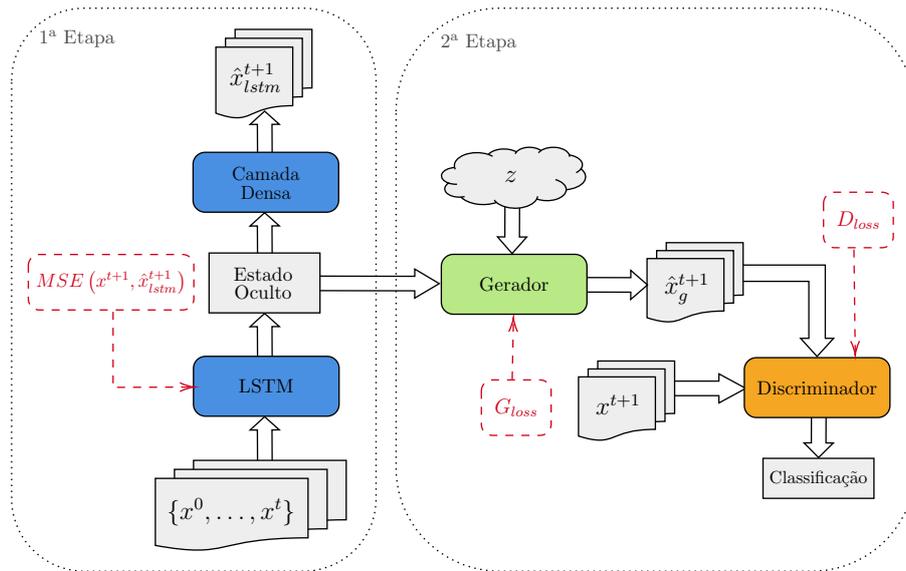


Figura 4.11: Modelo GAN condicionado por uma camada LSTM

vetor z passe a ser desconsiderado durante o treinamento. Isso ocorre uma vez que a rede é apresentada a mesma amostra diversas vezes durante o seu treinamento, cada uma com um valor de z diferente, o que faz com que a rede desconsidere a relevância do z para a formação daquela amostra, ao minimizar o MSE.

De forma a buscar mitigar estes efeitos e conseguir gerar amostras no mesmo instante de tempo que tenham a mesma estatística do dado real, foram propostas modificações neste modelo apresentado.

4.2.3 GAN de Recorrência Codificada

Para reduzir os efeitos apresentados a arquitetura do modelo foi alterada de forma a separar a rede do gerador em duas partes: a primeira consiste em uma LSTM que será treinada de forma a minimizar o MSE na estimação do próximo instante da série temporal; a segunda parte recebe a o estado oculto da LSTM e o utiliza como entrada combinado com o vetor z em uma MLP, treinada para minimizar apenas o erro adversário. Desta forma o uso do MSE não afeta os pesos posteriores a entrada do vetor z , de modo a garantir seu papel de promover variabilidade entre as amostras. O modelo foi chamado de GAN de Recorrência Codificada (Encoded Recurrence GAN - ER-GAN) e pode ser visualizado na Figura 4.11.

O modelo foi treinado de forma semelhante a do anterior, durante 1000 épocas com um conjunto de dados sintéticos de 5000 amostras, das quais 4000 foram utilizadas para treinamento e 1000 para teste. O treinamento foi feito usando Adam gradiente descendente minimizando, na primeira etapa o MSE da LSTM e na segunda o erro adversário da MLP do gradiente. Inicialmente foi testada uma taxa de

aprendizado de 0,001 e um tamanho de janela de 20 amostras apenas para verificar o comportamento do modelo. Pode-se observar que mesmo sem a calibração desses parâmetros o modelo já foi capaz de convergir para uma distribuição bastante fiel a dos dados sintéticos, como pode ser visto na Figura 4.12. Pode-se observar na Figura A.1 do Apêndice A, a evolução do treinamento nas primeiras épocas, mostrando que a partir da época 150 o modelo já apresentava uma distribuição semelhante a dos dados sintéticos.

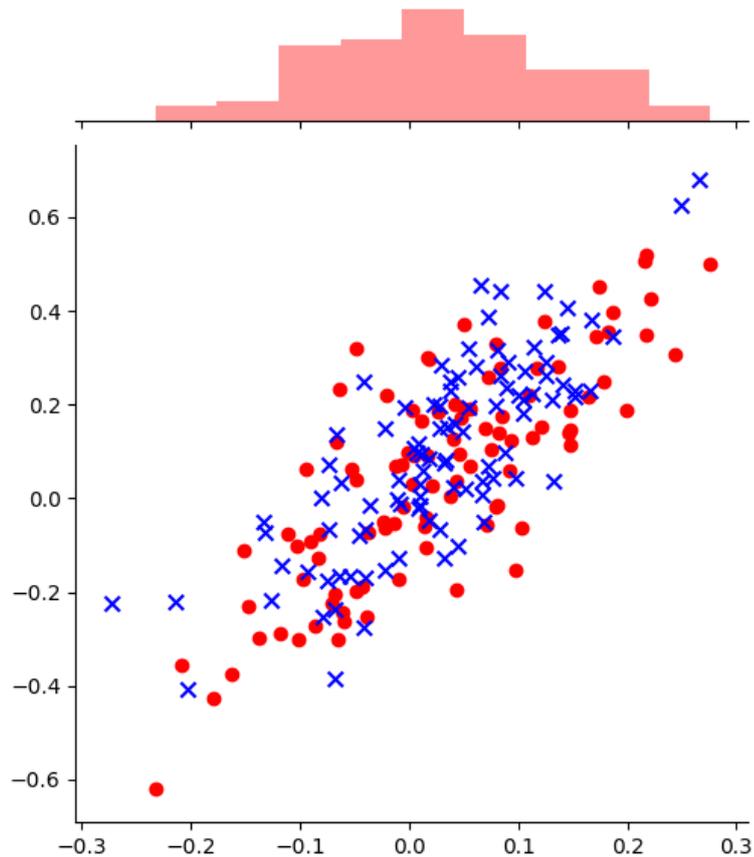


Figura 4.12: Distribuição dos dados gerados no instante ($t = 1$) (em vermelho) em contraste com a distribuição esperada para os dados sintéticos (em azul) usando o modelo de GAN condicionada com LSTM

A seguir, avaliou-se diferentes comprimentos de janela e taxas de aprendizado. Para cada valor avaliado foram treinados 50 modelos. Para cada um deles foram avaliados o MSE da estimação do valor esperado assim como a Norma de Frobenius da diferença entre as matrizes de covariância dos dados sintéticos e a calculada a partir dos dados gerados. A comparação entre os valores foi feita usando a média e o erro padrão dos resultados dos 50 modelos. Como observou-se que o modelo foi capaz de convergir rapidamente, reduziu-se o número de épocas de treinamento para 300, a fim de agilizar o processo de treinamento. Os comprimentos de janela avaliados

foram de 20, 50 e 100 amostras, comumente utilizados na análise técnica de ativos financeiros [82]. O resultado mostrado na Figura 4.13 mostra que o comprimento de janela de 20 amostras apresentou um resultado superior aos demais. Também foram avaliadas as taxas de aprendizado de 0,0001, 0,0005, 0,001, 0,005 e 0,01. Como pode ser visto na Figura 4.14, a taxa de 0.005 apresentou o melhor resultado dentre as taxas avaliadas.

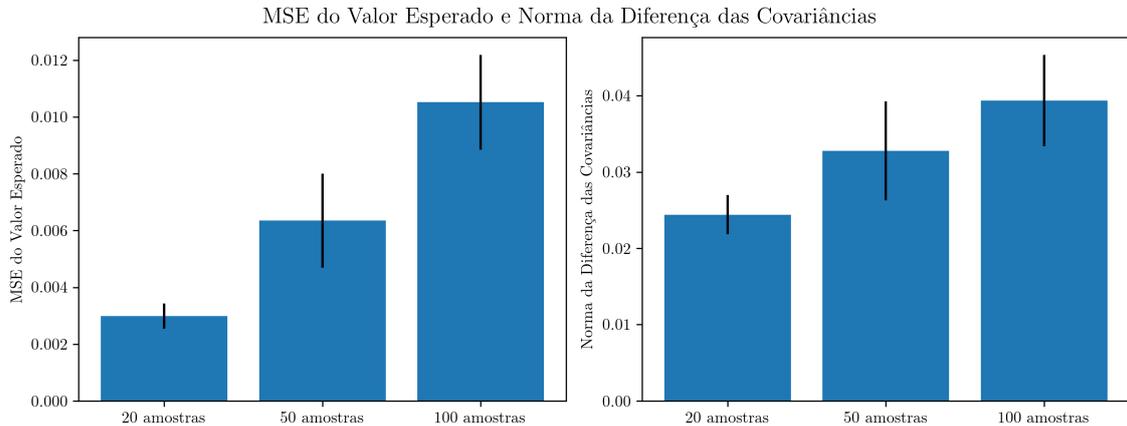


Figura 4.13: Comparação entre diferentes tamanhos de janela. Foram avaliadas janelas de 20, 50 e 100 amostras, comumente utilizados na análise técnica de ativos financeiros. Pode-se observar que a janela de 20 amostras apresentou o melhor resultado.

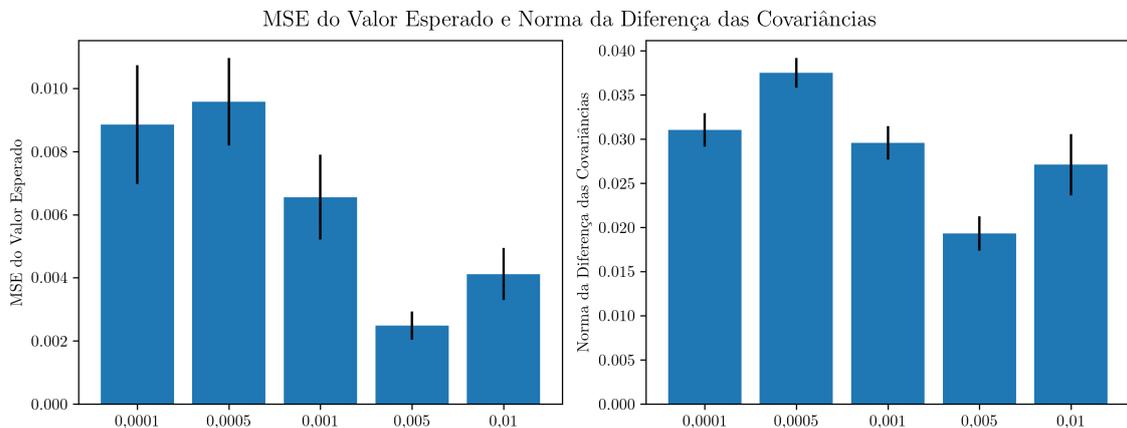


Figura 4.14: Comparação entre diferentes taxas de aprendizado. Foram utilizadas as taxas de aprendizado de 0,0001, 0,0005, 0,001, 0,005 e 0,01. Pode-se observar que a taxa de 0.005 apresentou melhor resultado.

Pode-se concluir que o modelo apresentado consegue superar as deficiências do modelo anterior. A seguir busca-se avaliar se a aplicação de técnicas apresentadas na Seção 3.3 pode melhorar os resultados da convergência da GAN.

4.2.4 Comparação entre Métodos de Seleção de Portfólio

Foi então feita uma comparação do modelo proposto com o método de Média-Variância descrito no Capítulo 5 e com um portfólio igualmente balanceado. Não foi feita nenhuma comparação com métodos de encolhimento uma vez que o conjunto de dados sintéticos engloba apenas dois ativos. Para fazer a comparação utilizou-se um as últimas 100 amostras das séries temporais para definir a média e covariância amostral do modelo MV, sendo então definido o portfólio que maximiza a Razão de Sharpe, descrita pela Equação 4.1.

$$SR = \frac{r_p - \mu_0}{\sigma_p} \quad (4.1)$$

A Razão de Sharpe é uma métrica usada para avaliar a relação entre risco e retorno. O portfólio tangente também é aquele que maximiza a Razão de Sharpe. Para o modelo de GAN proposto foram utilizadas as mesmas 100 amostras como entrada do modelo. A partir destas entradas, tendo implícita a distribuição de probabilidade do próximo instante de tempo, o modelo gerou 1000 amostras referentes a este próximo instante. Estas 1000 amostras, foram então utilizadas para o cálculo da média e covariância amostral e definindo-se o o portfólio que maximiza a Razão de Sharpe. Foi escolhida como taxa de retorno livre de risco para a Razão de Sharpe de 0.02, que é aproximadamente a taxa de um título da dívida dos EUA de 10 anos, comumente usada como taxa livre de risco. Este processo foi repetido por 100 instantes de tempo gerando o gráfico apresentado na Figura 4.15, que compara o retorno acumulado de cada um dos métodos.

Neste cenário a ER-GAN apresentou o mesmo resultado que o do modelo de média-variância, ambos superiores ao resultado do portfólio igualmente balanceado. Uma vez que o portfólio apresentado possui apenas dois ativos ambos os modelos alocaram 100% do capital no ativo de maior retorno e maior volatilidade, que, neste caso foi a configuração que maximizou a Razão de Sharpe.

Considerando os resultados deste capítulo a ER-GAN demonstrou ter um forte potencial para ser aplicado com dados de séries temporais financeiras reais. O modelo se mostrou capaz de reproduzir de forma fiel o comportamento estatístico dos dados sintéticos além de conseguir apresentar o mesmo comportamento do modelo de média-variância para um caso simples. Uma vez que o modelo média-variância tem que restringir o número de amostras usadas, pois o comportamento estatístico de retornos distantes pode variar muito e a ER-GAN a partir de amostras do passado consegue inferir a distribuição do próximo instante de tempo espera-se que para exemplos mais complexos com dados reais a ER-GAN possa apresentar resultados superiores.

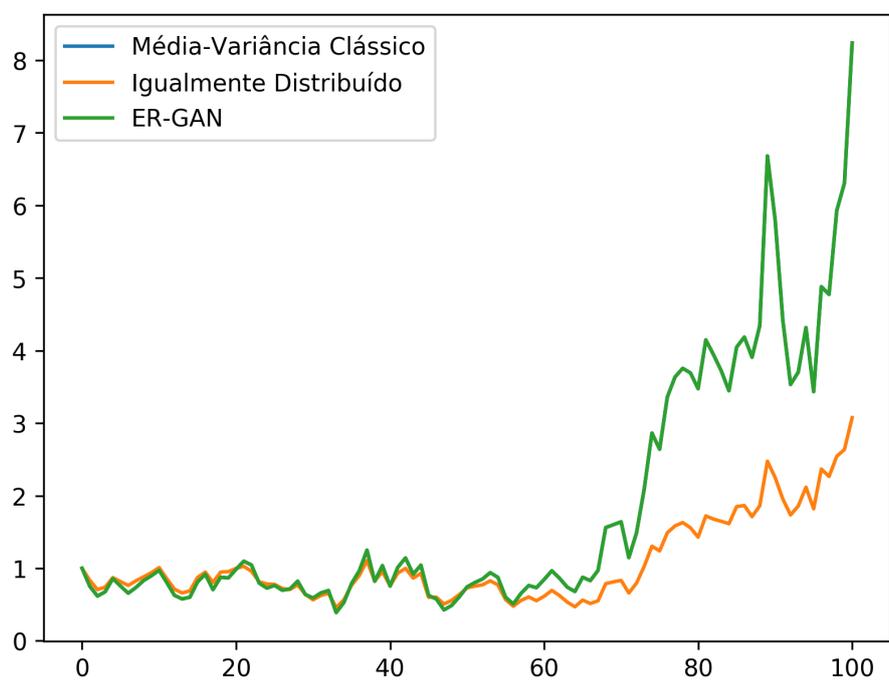


Figura 4.15: Comparação entre a ER-GAN, o modelo de Média-Variância e um portfólio igualmente balanceado

Capítulo 5

Aplicação a Dados de Ativos Reais

Neste capítulo o modelo apresentado no Capítulo 4 será aplicado a dados provenientes de séries temporais reais. Serão apresentados os conjuntos de dados utilizados assim como os processamentos feitos para gerar as variáveis utilizadas pelo modelo. O resultado do modelo treinado com cada um dos datasets é comparado a diferentes modelos utilizados para a seleção de portfólio já discutidos.

Uma vez avaliada a viabilidade da aplicação de GANs para a seleção de portfólios usando dados de ativos sintéticos, pode-se passar para a aplicação em dados reais. Este capítulo descreve a metodologia utilizada para o treinamento do modelo ER-GAN com dados reais. Inicialmente são descritos e analisados os datasets utilizados assim como o pré-processamento aplicado aos dados. A seguir é descrita a metodologia utilizada para chegar a melhor configuração para o modelo. São descritos os parâmetros que foram testados assim como as diferentes técnicas aplicadas no treinamento. Por fim são definidos os benchmarks para a comparação final dos resultados.

5.1 Dados e Variáveis Utilizados

Para este projeto foram utilizados dois datasets de dados de ativos da bolsa de valores dos EUA, um com uma quantidade reduzida de ativos (11 ativos) e um segundo com uma quantidade mais elevada (100 ativos), para avaliar o comportamento do modelo com o aumento de ativos no portfólio. A seguir é descrito em detalhes os datasets assim como é feita uma análise estatística de suas variáveis.

5.1.1 Um Ativo por Indústria

A complexidade de uma RNN, referente ao número de pesos a serem otimizados durante o treinamento, é diretamente associada a quantidade de variáveis utilizadas como entrada do modelo. Essa complexidade implica na necessidade por uma

quantidade maior de amostras para serem usadas durante o treinamento a fim de que o modelo tenha capacidade de generalização para amostras fora do conjunto de treinamento. Devido a isso optou-se por inicialmente utilizar um dataset com um conjunto mais restrito de ativos e em seguida avaliar o desempenho do modelo com um dataset com mais ativos.

O universo de ativos utilizado para a seleção de portfólios no primeiro dataset consiste em um total de 11 ativos cotados na bolsa dos Estados Unidos. Foi selecionada uma ação para cada um dos 11 setores da economia, definidos pelo Global Industry Classification Standard. Essa escolha foi tomada a fim de conseguir construir um dataset contendo ativos o mais descorrelacionados e diversos o possível, mesmo com um número relativamente baixo de ativos. A ação escolhida para cada setor foi aquela que possuía a maior capitalização de mercado no dia 24/01/2019 (momento em que os dados foram obtidos), uma vez que os ativos mais capitalizados costumam ser aqueles com maior liquidez de forma que transações individuais possuem pouco impacto nos preços dos ativos, reduzindo a possibilidade de amostras espúrias no conjunto. A lista completa dos ativos encontra-se no Apêndice B.

Foram utilizados dados do período que começa em 02/01/1998 e vai até 24/01/2019, sendo este o maior período disponível para os ativos em questão no provedor dos dados [83]. Este período apresenta um total de 5300 amostras para cada um dos ativos.

5.1.2 100 Ativos do S&P 500

Para construir um portfólio com um número maior de ativos optou-se por utilizar os ativos presentes no índice S&P 500, índice que contém as 500 empresas com maior capitalização na bolsa dos EUA. O motivo por optar por ativos de alta capitalização já foi descrito previamente. Como existem ativos no índice de empresas que foram listadas em bolsa recentemente optou-se por não utilizar todos os 500 ativos do índice, uma vez que a janela tempo disponível para o treinamento passaria a ficar limitada pelo ativo mais recente. Utilizou-se portanto o conjunto dos 100 ativos com maior número de amostras disponível, presentes no índice no dia 24/01/2019 (momento em que os dados foram obtidos). Desta forma foi possível formar um dataset diverso, com uma grande quantidade de ativos altamente capitalizados, mas sem impactar na quantidade de amostras disponíveis para treinamento. A lista completa dos ativos encontra-se no Apêndice C.

Foram utilizados dados do período que começa em 05/01/1998 e vai até 24/01/2019, sendo este o maior período disponível para os ativos em questão no provedor dos dados [83]. Este período apresenta um total de 5298 amostras para cada um dos ativos.

5.1.3 Análise das Variáveis

Cada amostra contida nos datasets contém as seguintes variáveis:

- Preço na abertura
- Preço no fechamento
- Preço máximo no dia
- Preço mínimo no dia
- Volume negociado
- Fechamento Ajustado
- Coeficiente de Split
- Dividendos distribuídos

Primeiramente o preço de abertura, fechamento, máxima e mínima foram ajustados com relação a splits e pagamentos de dividendos. Estes são eventos que ocorrem esporadicamente e alteram e devem ser refletidos em um ajuste no preço dos ativos. A seguir as variáveis foram combinadas de forma a gerar novas variáveis, em forma de razões. Isto é feito para evitar a necessidade de normalização e não afetar o treinamento do modelo. As variáveis criadas foram as seguintes:

- Retorno
- Razão fechamento/abertura (F/A)
- Razão fechamento/máxima (F/Max)
- Razão fechamento/mínima (F/Min)
- Variação do volume (VV)

Desta forma o modelo passa a ser composto por um total de 55 variáveis de entrada (cinco variáveis para cada uma das 11 ações) e 11 saídas(uma para cada ação).

Uma vez aplicados os pré-processamentos iniciais foi feita uma análise nas variáveis geradas. Primeiramente observou-se sua distribuição em quartis de forma a verificar a ordem de grandeza de cada variável. Pode-se observar que o valor da variação de volume consegue atingir faixas com valor absoluto consideravelmente superior as demais. Optou-se por escalar a variável de variação de treinamento, dividindo-a por 100, buscando coloca-la numa faixa mais próxima das demais.

Tabela 5.1: A tabela mostra a média o desvio padrão e os limites dos quartis da distribuição de cada uma das variáveis, considerando todos os ativos.

	Retorno	F/A	F/Max	F/Min	VV
Média	0,000247	-0,000027	-0,011420	0,011967	-0,083941
Desvio Padrão	0,021207	0,018428	0,014304	0,013976	0,831129
Mínimo	-0,548575	-0,414286	-0,428571	0,000000	-149,000000
25%	-0,008052	-0,007449	-0,014459	0,003893	-0,257575
50%	0,000387	0,000000	-0,007384	0,008093	-0,012215
75%	0,008900	0,007763	-0,003303	0,015047	0,192181
Máximo	0,257912	0,252475	0,000000	0,252475	0,987291

A seguir avaliou-se a distribuição conjunta dos retornos de cada um dos ativos no universo do portfólio, conforme pode ser visto na Figura 5.1. Pode-se observar claramente que alguns ativos possuem uma correlação entre si maior do que outros. Isso evidencia a necessidade de um bom estimador para a matriz de covariâncias e seu impacto na construção de um bom portfólio, conforme foi descrito de forma teórica no Capítulo 5.

Observou-se também o comportamento dos retornos e variâncias ao longo do tempo. Para isso as séries temporais dos retornos foram separadas em janelas de 200 amostras. Para cada janela calculou-se a média e a variância dos retornos durante este período de 200 amostras. O modelo de média-variância original utiliza da premissa de que é possível modelar o valor esperado e covariância no próximo instante por meio da média e covariância amostrais. Para esta hipótese ser válida, seria esperado ver os valores calculados em cada uma das janelas se manter aproximadamente constante. O que pode ser visto, porém, na Figura 5.2 é que tanto a média quanto a variância apresentam variações consideráveis ao longo do tempo. Devido a isso, quem utiliza o modelo de média-variância na prática, costuma utilizar janelas das séries temporais, conforme já foi discutido no Capítulo 5.

Para ter uma visualização ainda mais clara da variação das distribuições ao longo do tempo a Figura 5.3 mostra a distribuição de um dos ativos do portfólio (AMZN) em dois períodos diferentes. Pode-se observar uma diferença considerável na cauda da distribuição. Espera-se então que o modelo GAN seja capaz de utilizar a informação de cada instante de tempo no passado, ponderando sua contribuição para a distribuição do próximo instante.

5.2 Metodologia do Experimento

A seguir, serão descritas diferentes configurações avaliadas para o modelo ER-GAN. Para cada configuração foi feito o mesmo processo de treinamento para avaliar a me-

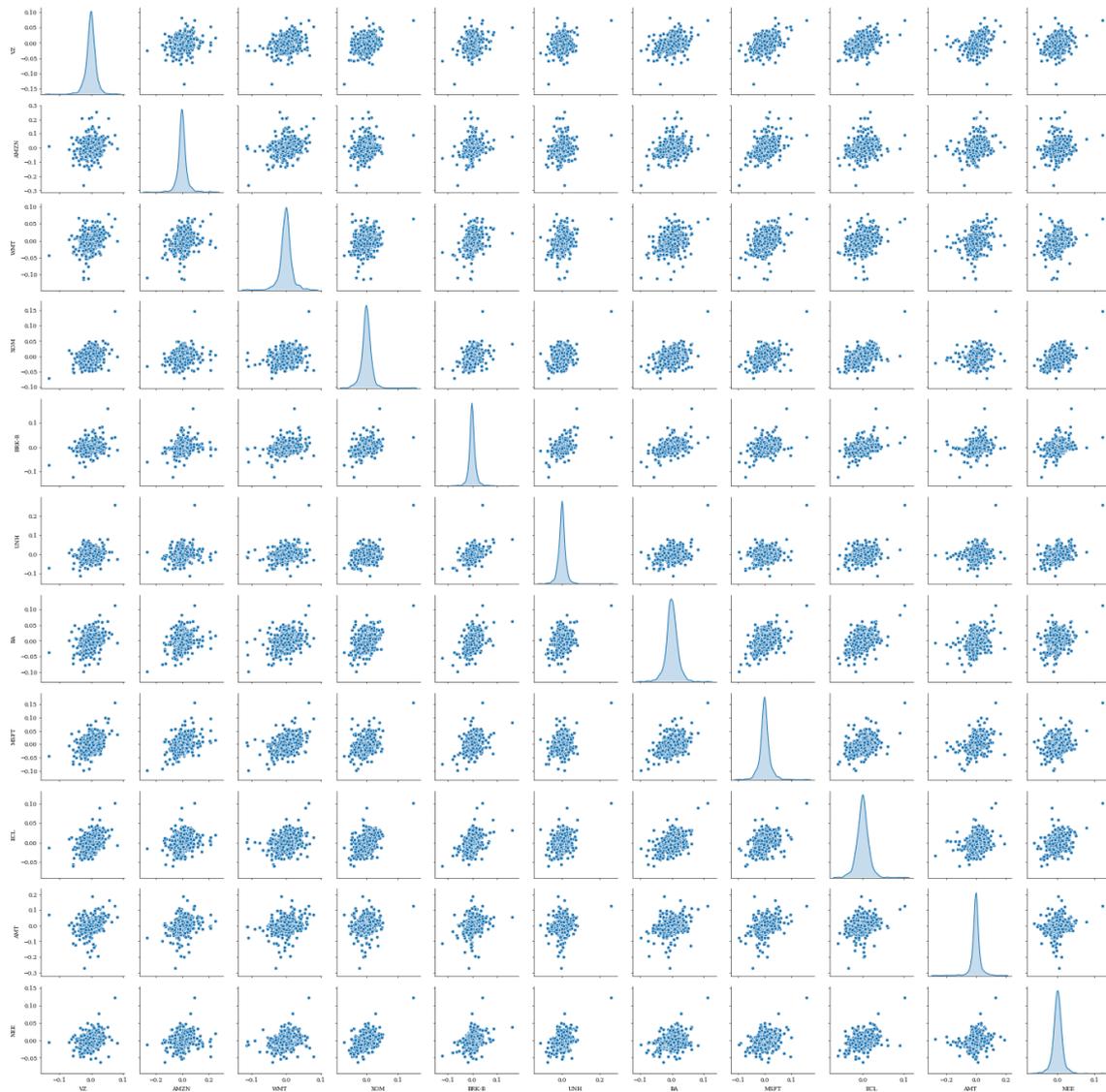


Figura 5.1: Distribuição conjunta dos retornos dos ativos presentes no portfólio

lhor configuração. Os dados foram inicialmente separados em conjuntos de treino, validação e teste, sendo 70% para treino, 20% para validação e 10% para teste. Para cada configuração foram treinados 50 modelos utilizando o conjunto de teste. Os treinamentos foram feitos por 300 épocas para minimizar o MSE usando Estimativa de Momento Adaptativo (Adaptive Moment Estimation - Adam) [78]. Após o treinamento foram feitas 200 simulações com dados do conjunto de validação para cada um dos 50 modelos. Para cada modelo foi calculado o valor de retorno médio, a variância do retorno e a Razão de Sharpe do modelo dentro do conjunto das 200 simulações. Foi então calculado um valor médio do retorno médio, da variância e da Razão de Sharpe dos 50 modelos, assim como seus respectivos erros padrão. O valor médio de Razão de Sharpe da configuração foi, então, usado para comparação entre diferentes configurações.

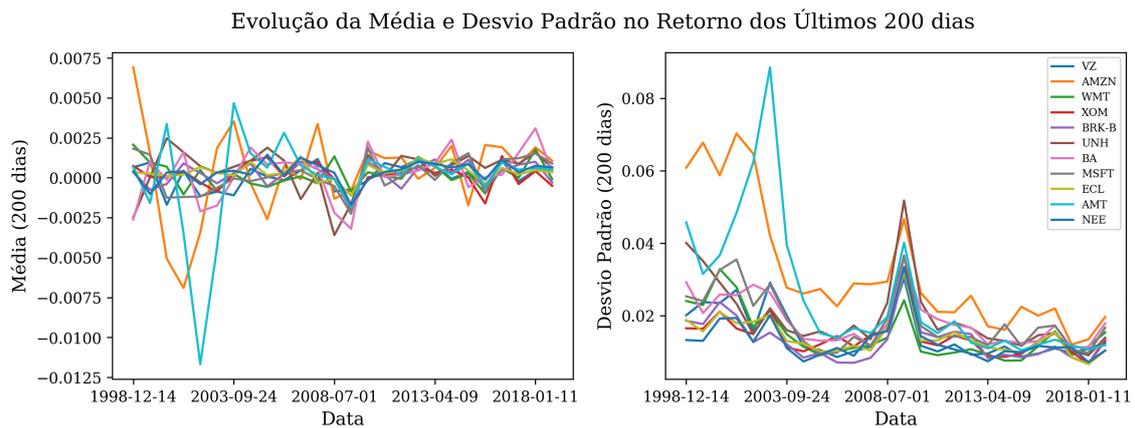


Figura 5.2: Evolução da média (esquerda) e da variância (direita) em janelas de 200 amostras. Tanto a média quanto a variância apresentam variações consideráveis ao longo do tempo.

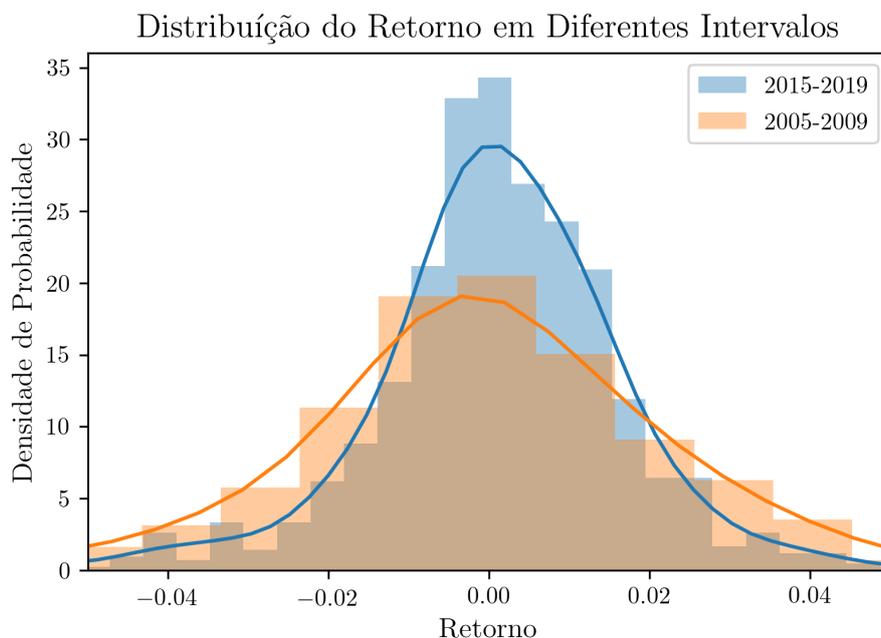


Figura 5.3: Distribuição de um ativo do portfólio em diferentes janelas de tempo. Pode-se observar uma diferença considerável na calda da distribuição.

5.2.1 Ajuste de Hiperparâmetros

A primeira etapa do experimento consistiu em definir os melhores hiperparâmetros para o modelo. Os seguintes hiperparâmetros foram avaliados:

- Numero de camadas escondidas LSTM no gerador
- Numero de hidden states nas camada LSTM
- Número de camadas na MLP no gerador

- Número de neurônios por camada MLP do gerador
- Número de camadas na MLP no discriminador
- Número de neurônios por camada MLP do discriminador
- Taxa de aprendizado do gerador
- Taxa de aprendizado do discriminador
- Tamanho da janela utilizado no treinamento da LSTM

Para os parâmetros referentes a topologia das redes foi feita uma busca aleatória avaliando 20 combinações aleatórias de possíveis topologias. A Tabela 5.2 mostra as 20 configurações avaliadas e seus respectivos valores para a quantidade de camadas LSTM, número de hidden units por camada LSTM, número de camadas e neurônios na MLP no gerador e número de camadas e neurônios na MLP no discriminador. Cada topologia foi treinada conforme descrito na Seção 5.2, sendo escolhida a configuração com maior Razão de Sharpe médio entre as 20.

Tabela 5.2: Configurações de topologia avaliadas por busca aleatória.

	LSTM	Gen. MLP	Disc. MLP
Conf. 1	2×10	20, 10	10
Conf. 2	3×20	10	10
Conf. 3	1×10	20	10, 50
Conf. 4	2×20	20	10, 50
Conf. 5	1×10	10	100
Conf. 6	2×50	20	50
Conf. 7	1×100	20, 10	10, 50
Conf. 8	3×20	50, 50	20, 10
Conf. 9	1×10	20, 10	20, 50, 100
Conf. 10	1×100	10, 50	10, 50
Conf. 11	1×20	100	50, 50
Conf. 12	3×10	20, 50, 100	50
Conf. 13	1×50	50, 50	10, 100
Conf. 14	3×50	10, 50	50
Conf. 15	3×10	50, 20, 100	50, 50
Conf. 16	2×20	50, 50	20, 50, 100
Conf. 17	1×50	50, 50	50, 20, 100
Conf. 18	2×100	50	50, 50
Conf. 19	3×100	10, 50	100
Conf. 20	2×100	50, 20, 100	50, 20, 100

A seguir foram avaliados diferentes valores para a taxa de aprendizado e para o tamanho da janela para esta configuração. Os valores avaliados foram os mesmos

utilizados no Capítulo 4. Avaliou-se as taxas de aprendizado 0,0001, 0,0005, 0,001, 0,005 e 0,01 e os tamanhos de janela de 20, 50 e 100 amostras. Novamente foram escolhidos os valores que apresentaram a maior Razão de Sharpe média.

5.2.2 Avaliação de Técnicas para o Treinamento da GAN

Na Seção 3.3 foram discutidas diferentes técnicas propostas para auxiliar na estabilidade do treinamento de GANs. Nenhuma delas apresenta os melhores resultados para todos os casos, conforme já foi discutido, de forma que faz-se necessário a avaliação destas para o problema em questão. Foi testada a técnica tradicional de treinamento para GANs, baseada em reduzir a distância de Jensen-Shannon assim como as variantes WGAN e WGAN-GP, que se baseiam na redução da distância de Wasserstein. Foi escolhida a técnica que apresentou o maior valor médio da Razão de Sharpe. Foi escolhida como taxa anual de retorno livre de risco para a Razão de Sharpe de 0.02, que é aproximadamente a taxa de um título da dívida dos EUA de 10 anos, comumente usada como taxa livre de risco. Este processo foi repetido por 100 instantes de tempo gerando o gráfico apresentado na Figura 4.15, que compara o retorno acumulado de cada um dos métodos.

Por fim, foram avaliadas as técnicas de discriminação de mini-lotes (Minibatch Discrimination - MB Disc) e normalização por mini-lotes (Batch Normalization - B Norm). Ambas as técnicas foram discutidas na Seção 3.3 e tem como função evitar o colapso de modo e aumentar a diversidade entre as amostra. Apesar de, segundo Salimans et al. [66], a discriminação por mini-lotes ser mais efetiva do que a normalização por mini-lotes, a aplicação da técnica foi popularizada por ter sido utilizada na DCGAN [62], que é um dos principais benchmarks de modelo generativo para imagens. Levando isso em conta decidiu-se avaliar se impacto de ambas no treinamento do modelo.

5.2.3 Comparação entre Métodos de Seleção de Portfólio

Por fim, 50 modelos foram treinados usando as melhores configurações de topologia, taxa de aprendizado, tamanho da janela, método de treinamento e técnica para reduzir o colapso do modo. O valor médio da Razão de Sharpe desta configuração foi então comparado com o de todas as outras combinações avaliadas. Por fim, foi selecionada aquela que apresentou a maior Razão Sharpe média no conjunto de validação. Entre os 50 modelos treinados da melhor configuração, o que apresentou o maior retorno médio na simulação de 200 amostras foi selecionado e comparado com diversos benchmarks benchmarks. Os benchmarks incluem a abordagem clássica de variância média, um portfólio igualmente balanceado e várias técnicas de encolhimento

- **Portfólio igualmente distribuído** (Equally Weighted - EW) - Possui o mesmo peso para cada um dos ativos. É comumente considerado como benchmark, pois devido a dificuldade de se ter uma boa estimativa para a covariância e valor esperado dos ativos em muitos casos um portfólio igualmente balanceado apresenta resultados equivalentes a portfólios calculados por modelos de otimização de portfólio, como demonstrado por DeMiguel et al. [3].
- **Modelo de média-variância** (Mean-Variance - MV) - Modelo clássico de média-variância utilizando a média amostral como aproximador para o valor esperado dos ativos e a covariância amostral como aproximador para a covariância.
- **Encolhimento para Fator Único** (Single Factor Shrinkage - SFS) - Proposto por Ledoit et al. [25], utiliza uma média ponderada da matriz de correlação amostral e da matriz de covariância do modelo de índice único de Sharpe como técnica de encolhimento.
- **Encolhimento para Correlação Constante** (Constant Correlation Shrinkage - CCS) - Proposto por Ledoit et al. [26], o encolhimento ajusta todos os pares de correlações para a correlação média e mantém as variâncias inalteradas.
- **Encolhimento para Variância Constante** (Constant Variance Shrinkage - CVS) - Proposto por Ledoit et al. [27], utiliza como alvo do encolhimento a matriz diagonal com a média das variâncias ativos nas diagonais e zeros em outro lugar.
- **Encolhimento por Aproximação Oráculo** (Oracle Approximating Shrinkage - OAS) - Proposto por Chen et al. [28] esta técnica de encolhimento apresenta um menor MSE do que o Encolhimento para Variância Constante para o caso de distribuições Gaussianas ou Semi-Gaussianas.

Para avaliar o desempenho do modelo ER-GAN com os benchmarks foram utilizados os dados do conjunto de teste. Cada modelo foi apresentado a uma amostra contendo os 100 dias, a fim de selecionar o melhor portfólio para o próximo dia. O processo foi repetido para rebalancear o portfólio diariamente por um período de 300 dias. A quantidade de amostras disponíveis para os modelos aumentou conforme o passar do tempo, de forma que para estimar o portfólio do dia 300 os modelos tiveram a disposição os 100 dias iniciais mais os 299 dias anteriores ao dia 300. Foram avaliados o retorno médio, a variância e a Razão de Sharpe do modelo ER-GAN e de cada um dos benchmarks.

5.3 Resultados e Discussões

5.3.1 Um Ativo por Indústria

Inicialmente foram treinados modelos para avaliar as 20 configurações de topologias selecionadas na busca aleatória. A Tabela 5.3 e a Figura 5.4 mostram os resultados das 20 configurações avaliadas para retorno médio, variância e Razão de Sharpe em ordem decrescente de Razão de Sharpe. A melhor configuração apresentou uma camada LSTM com 100 hidden units como encoder, 2 camadas totalmente conexas no gerador, com 10 e 50 neurônios respectivamente e duas camadas totalmente conexas no discriminador com 10 e 50 neurônios respectivamente.

Tabela 5.3: Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN. A tabela mostra em ordem decrescente de Razão de Sharpe. A melhor configuração apresentou uma camada LSTM com 100 hidden units como encoder, 2 camadas totalmente conexas no gerador, com 10 e 50 neurônios respectivamente e duas camadas totalmente conexas no discriminador com 10 e 50 neurônios respectivamente.

	Média $\times 10^{-4}$	Variância $\times 10^{-2}$	SR $\times 10^{-2}$
Conf. 10	7,477 \pm 0,315	1,050 \pm 0,014	6,350 \pm 0,270
Conf. 13	6,949 \pm 0,334	1,039 \pm 0,018	5,901 \pm 0,273
Conf. 7	7,020 \pm 0,308	1,056 \pm 0,017	5,867 \pm 0,262
Conf. 2	6,625 \pm 0,250	1,014 \pm 0,014	5,764 \pm 0,236
Conf. 20	6,783 \pm 0,319	1,046 \pm 0,014	5,704 \pm 0,281
Conf. 3	6,622 \pm 0,260	1,028 \pm 0,015	5,651 \pm 0,237
Conf. 8	6,626 \pm 0,273	1,040 \pm 0,015	5,625 \pm 0,245
Conf. 17	6,693 \pm 0,337	1,056 \pm 0,021	5,555 \pm 0,261
Conf. 4	6,503 \pm 0,319	1,052 \pm 0,019	5,528 \pm 0,290
Conf. 11	6,652 \pm 0,329	1,064 \pm 0,016	5,464 \pm 0,285
Conf. 18	7,129 \pm 0,493	1,159 \pm 0,032	5,444 \pm 0,338
Conf. 5	6,998 \pm 0,456	1,138 \pm 0,033	5,417 \pm 0,320
Conf. 6	6,542 \pm 0,318	1,052 \pm 0,016	5,410 \pm 0,247
Conf. 12	6,355 \pm 0,269	1,028 \pm 0,015	5,404 \pm 0,237
Conf. 1	6,217 \pm 0,272	1,030 \pm 0,015	5,349 \pm 0,270
Conf. 9	6,464 \pm 0,340	1,085 \pm 0,018	5,340 \pm 0,319
Conf. 16	6,435 \pm 0,359	1,063 \pm 0,015	5,289 \pm 0,312
Conf. 19	6,104 \pm 0,280	1,020 \pm 0,015	5,246 \pm 0,265
Conf. 14	6,220 \pm 0,317	1,054 \pm 0,018	5,221 \pm 0,298
Conf. 15	5,634 \pm 0,355	1,078 \pm 0,023	4,504 \pm 0,287

A Figura 5.5 mostra distribuição conjunta entre os retornos de dois dos ativos do portfólio. Sua evolução ao longo do treinamento pode ser vista na Figura A.1 do Apêndice A. São comparadas amostras geradas para o próximo instante de tempo com amostras em comparação com uma distribuição amostrada previamente a partir

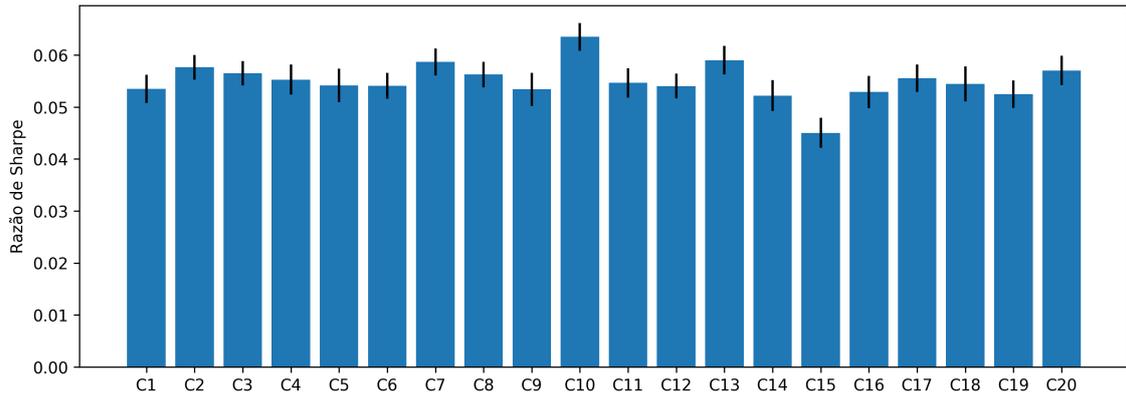


Figura 5.4: Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN.

dos valores de média e covariância destes ativos. É possível ver que as amostras geradas para o próximo instante de tempo apresentam uma grande sobreposição àquelas da distribuição original. A distribuição indica que o modelo não incorreu em colapso de modo, pois as amostras não se apresentaram concentradas próximas a um único ponto.

A seguir foram avaliadas diferentes taxas de aprendizado e tamanho da janela usada no treinamento da rede LSTM. A Tabela 5.4 e na Figura 5.6 mostram a comparação das diferentes taxas de aprendizado. O valor de taxa de aprendizado que apresentou o maior valor de razão de sharpe foi de 0,0001.

Tabela 5.4: Comparação entre taxas de aprendizado para o dataset Um Ativo por Indústria.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	Razão de Sharpe ($\times 10^{-2}$)
0,0001	$7,564 \pm 0,288$	$1,026 \pm 0,014$	$6,583 \pm 0,242$
0,0005	$7,081 \pm 0,278$	$1,042 \pm 0,016$	$6,053 \pm 0,253$
0,001	$6,416 \pm 0,256$	$1,022 \pm 0,012$	$5,511 \pm 0,252$
0,005	$7,477 \pm 0,315$	$1,050 \pm 0,014$	$6,350 \pm 0,270$
0,01	$6,360 \pm 0,298$	$1,045 \pm 0,018$	$5,367 \pm 0,265$

Já o melhor tamanho de janela manteve-se sendo de 20 amostras, valor usado durante o treinamento das diferentes topologias, como pode ser visto na Tabela 5.5 e na Figura 5.7.

A seguir comparou-se o tipo de treinamento utilizado pela GAN, novamente avaliando o treinamento padrão, o modelo WGAN e o WGAN-GP, tanto no retorno acumulado quanto em relação ao Razão de Sharpe, como pode ser visto na Tabela 5.6 e na Figura 5.8.

Por último avaliou-se o uso de discriminação de mini-lotes e normalização por

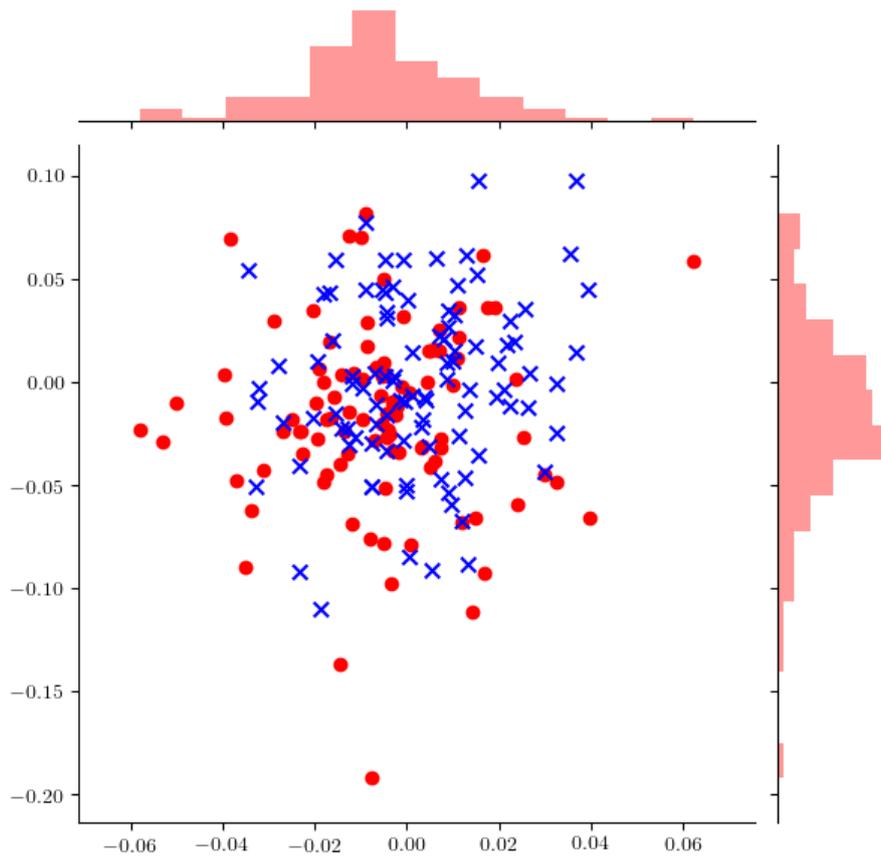


Figura 5.5: Distribuição conjunta real e gerada pela ER-GAN de dois ativos do portfólio. Pode-se observar que a distribuição da ER-GAN gera amostras menos dispersas, mas dentro da distribuição original, conforme o esperado.

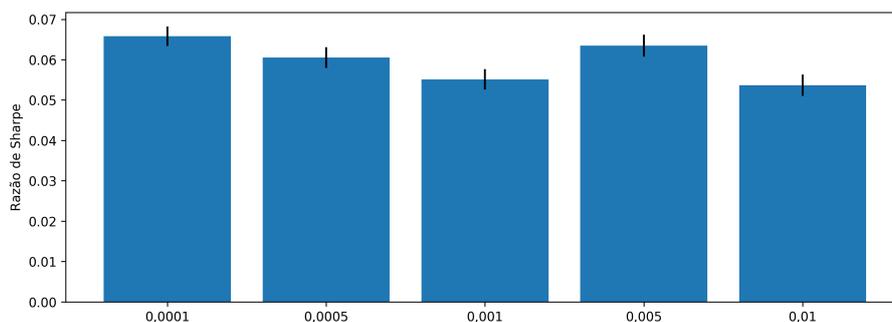


Figura 5.6: Comparação entre taxas de aprendizado para o dataset Um Ativo por Indústria.

mini-lotes. O modelo ER-GAN sem nenhuma das técnicas apresentou resultado superior, tanto em retorno acumulado e do ponto de vista do Razão de Sharpe, como pode ser visto na Tabela 5.7 e na Figura 5.9. Este resultado confirma que o modelo GAN padrão, não aparenta sofrer de colapso de modo. Isso confirma o que pode ser observado tanto na Figura 4.12 quanto na Figura 5.5, onde as amostras dos

Tabela 5.5: Comparação entre tamanhos de janela para o dataset Um Ativo por Indústria.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
20	7,477 \pm 0,315	1,050 \pm 0,014	6,350 \pm 0,270
50	7,234 \pm 0,356	1,072 \pm 0,020	5,953 \pm 0,278
100	7,102 \pm 0,296	1,062 \pm 0,018	6,027 \pm 0,266

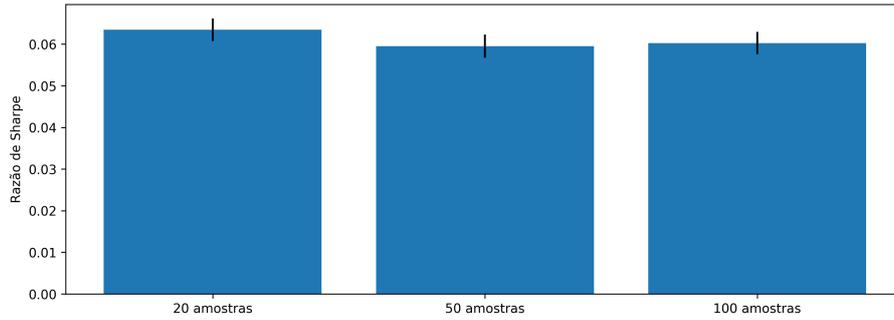


Figura 5.7: Comparação entre tamanhos de janela para o dataset Um Ativo por Indústria.

Tabela 5.6: Comparação entre métodos de treinamento de GAN para o dataset Um Ativo por Indústria.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
GAN	7,477 \pm 0,315	1,050 \pm 0,014	6,350 \pm 0,270
WGAN	6,039 \pm 0,389	1,049 \pm 0,016	4,998 \pm 0,342
WGAN-GP	7,213 \pm 0,348	1,034 \pm 0,026	6,298 \pm 0,313

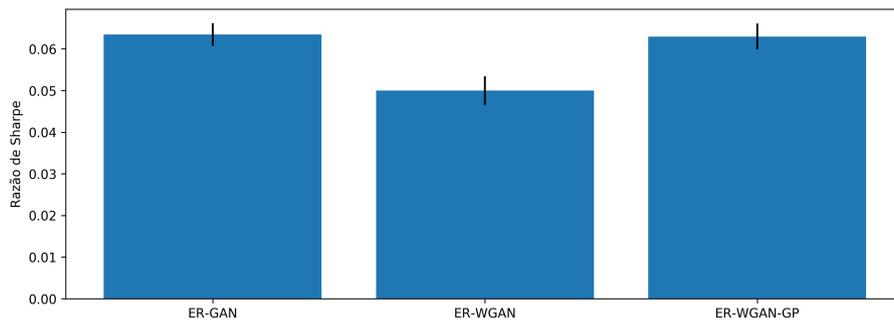


Figura 5.8: Comparação entre métodos de treinamento de GAN para o dataset Um Ativo por Indústria.

dados gerados se apresentaram bem distribuídas em comparação aos dados reais.

Em seguida, 50 modelos foram treinados combinando as melhores configurações apresentadas. O valor médio da Razão de Sharpe desse modelo nas amostras de

Tabela 5.7: Comparação entre métodos para a melhoria da diversidade das amostras com o dataset Um Ativo por Indústria.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
GAN	$7,477 \pm 0,315$	$1,050 \pm 0,014$	$6,350 \pm 0,270$
MB Disc,	$7,058 \pm 0,309$	$1,071 \pm 0,016$	$5,823 \pm 0,257$
B Norm,	$6,016 \pm 0,355$	$1,045 \pm 0,017$	$5,037 \pm 0,329$

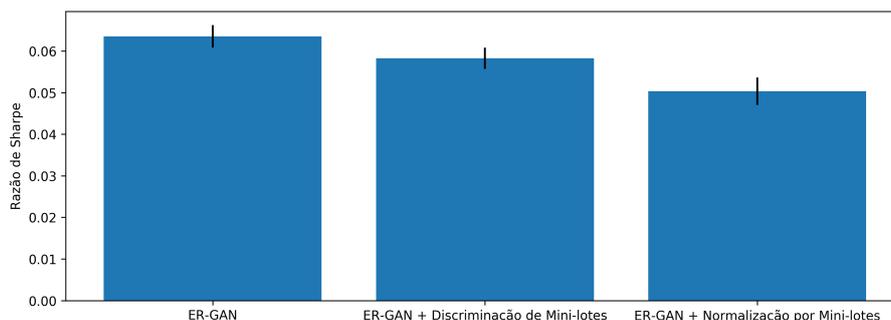


Figura 5.9: Comparação do retorno acumulado entre métodos para a melhoria da diversidade das amostras.

validação foi de $(7.100 \pm 0,227) \times 10^{-2}$, superior ao resultado de qualquer outra configuração independente. Em seguida, foi selecionado o modelo que apresentou o maior retorno médio entre os 50 e este foi usado para comparar o modelo com os benchmarks, agora usando o conjunto de testes. O resultado da média, variância e Razão de Sharpe pode ser visto na Tabela 5.8. O modelo ER-GAN apresentou um valor médio de retorno mais alto e uma Razão de Sharpe maior que todos os outros benchmarks. Seu retorno médio foi 14,11% maior que o benchmark com maior retorno e sua Razão de Sharpe foi 2,62% maior que o melhor benchmark.

como pode ser visto na Figura 5.10 quanto em relação ao retorno médio e Razão de Sharpe, como pode ser visto na Tabela 5.8.

Tabela 5.8: Comparação entre o modelo ER-GAN e benchmarks para o dataset Um Ativo por Indústria.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
MV	0.277	0.158	1.744
EW	0.303	0.114	2.661
SFS	0.290	0.161	1.796
CCS	0.302	0.164	1.837
CVS	0.325	0.157	2.070
OAS	0.305	0.158	1.921
ER-GAN	0.371	0.135	2.731

O retorno acumulado do modelo ER-GAN pode ser visto em Figura 5.10, e evidencia a diferença entre o desempenho dos diferentes modelos, com o modelo ER-GAN claramente superando os benchmarks. Pode-se observar também que o modelo ER-GAN apresentou menor sensibilidade a forte queda que aconteceu entre os instantes 200 e 250.

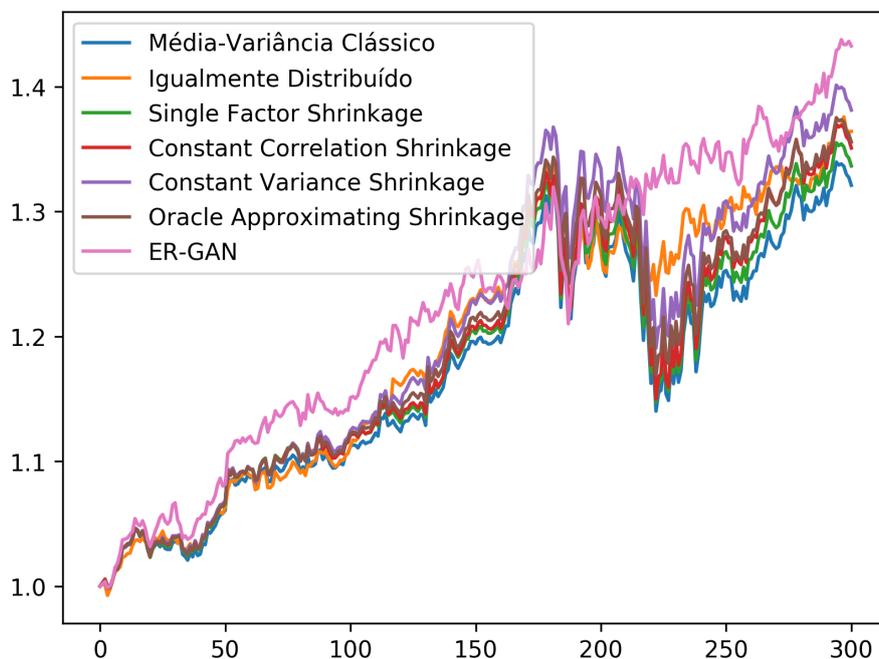


Figura 5.10: Comparação do retorno acumulado entre a melhor configuração de ER-GAN e os benchmarks. Pode-se observar que o portfólio que utiliza a ER-GAN apresentou um resultado substancialmente superior ao demais. A distância para o segundo melhor foi consideravelmente superior a obtida na configuração utilizada com os dados sintéticos.

5.3.2 100 Ativos do S&P 500

Uma vez que a ER-GAN apresentou um resultado satisfatório com o universo de apenas 11 ativos, o mesmo experimento foi realizado com um portfólio composto pelo dataset dos 100 ativos de maior capitalização presentes no S&P 500, de forma a verificar se o modelo ER-GAN é robusto a variações no tamanho do portfólio. O aumento de ativos, como já foi mencionado, impacta diretamente na complexidade do modelo e no número de parâmetros que são necessários serem otimizados via gradiente descendente. Porém os benchmarks também são afetados por um aumento no universo de ativos no portfólio, já que a quantidade de amostras utilizadas para a estimativa da matriz de covariâncias começa a não ser suficiente para uma estimativa precisa.

Novamente foram treinados modelos para avaliar as 20 configurações de topologias selecionadas na busca aleatória. A Tabela 5.9 e a Figura 5.11 mostram os resultados das 20 configurações avaliadas para retorno médio, variância e Razão de Sharpe em ordem decrescente de Razão de Sharpe. A melhor configuração apresentou duas camadas LSTM com 20 hidden units como encoder, 2 camadas totalmente conexas de 50 neurônios no gerador e três camadas totalmente conexas no discriminador com 20, 50 e 100 neurônios respectivamente.

Tabela 5.9: Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN com o dataset de 100 Ativos do S&P 500. A tabela mostra em ordem decrescente de Razão de Sharpe. A melhor configuração apresentou duas camadas LSTM com 20 hidden units como encoder, 2 camadas totalmente conexas de 50 neurônios no gerador e três camadas totalmente conexas no discriminador com 20, 50 e 100 neurônios respectivamente.

	Média $\times 10^{-4}$	Variância $\times 10^{-2}$	SR $\times 10^{-2}$
Conf. 16	7,878 \pm 0.419	0,925 \pm 0,016	7,721 \pm 0,460
Conf. 6	7,739 \pm 0.440	0,914 \pm 0,013	7,574 \pm 0,441
Conf. 10	7,380 \pm 0.410	0,891 \pm 0,011	7,412 \pm 0,444
Conf. 3	7,551 \pm 0.151	0,920 \pm 0,007	7,374 \pm 0,184
Conf. 2	7,559 \pm 0.125	0,921 \pm 0,004	7,347 \pm 0,135
Conf. 13	7,353 \pm 0.358	0,908 \pm 0,011	7,271 \pm 0,399
Conf. 18	7,033 \pm 0.319	0,884 \pm 0,009	7,113 \pm 0,370
Conf. 4	7,212 \pm 0.181	0,908 \pm 0,005	7,075 \pm 0,204
Conf. 11	7,014 \pm 0.327	0,897 \pm 0,010	7,039 \pm 0,393
Conf. 19	7,198 \pm 0.550	0,960 \pm 0,023	6,929 \pm 0,531
Conf. 12	7,732 \pm 0.587	1,025 \pm 0,025	6,723 \pm 0,476
Conf. 14	6,985 \pm 0.554	0,963 \pm 0,023	6,647 \pm 0,576
Conf. 7	6,581 \pm 0.356	0,897 \pm 0,013	6,566 \pm 0,414
Conf. 17	6,775 \pm 0.474	0,940 \pm 0,017	6,548 \pm 0,522
Conf. 8	6,710 \pm 0.438	0,923 \pm 0,012	6,510 \pm 0,477
Conf. 1	6,617 \pm 0.402	0,927 \pm 0,013	6,441 \pm 0,450
Conf. 9	6,397 \pm 0.544	0,922 \pm 0,018	6,341 \pm 0,563
Conf. 20	6,573 \pm 0.513	0,965 \pm 0,015	6,118 \pm 0,539
Conf. 15	6,297 \pm 0.496	0,955 \pm 0,019	5,911 \pm 0,541
Conf. 5	5,807 \pm 0.568	1,020 \pm 0,021	5,218 \pm 0,541

A Figura 5.5 mostra que as distribuições do dado real e do gerado pelo gerador continuam apresentando uma grande sobreposição, assim como apresentavam para o dataset de 10 ativos. Mais uma vez a distribuição indica que o modelo não incorreu em colapso de modo.

A seguir foram avaliadas diferentes taxas de aprendizado e tamanho da janela usada no treinamento da rede LSTM. A Tabela 5.4 e na Figura 5.6 mostram a comparação das diferentes taxas de aprendizado. O valor de taxa de aprendizado

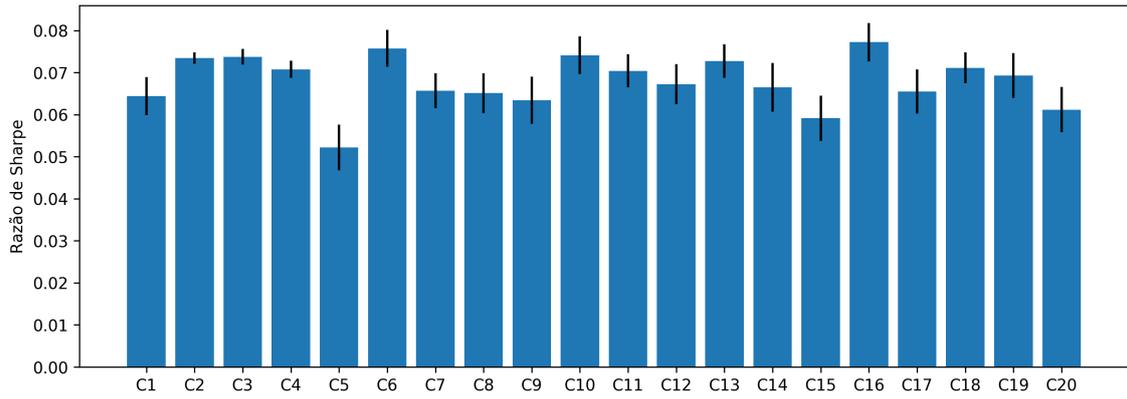


Figura 5.11: Comparação entre diferentes configurações para as topologias das redes do modelo ER-GAN com o dataset de 100 Ativos do S&P 500

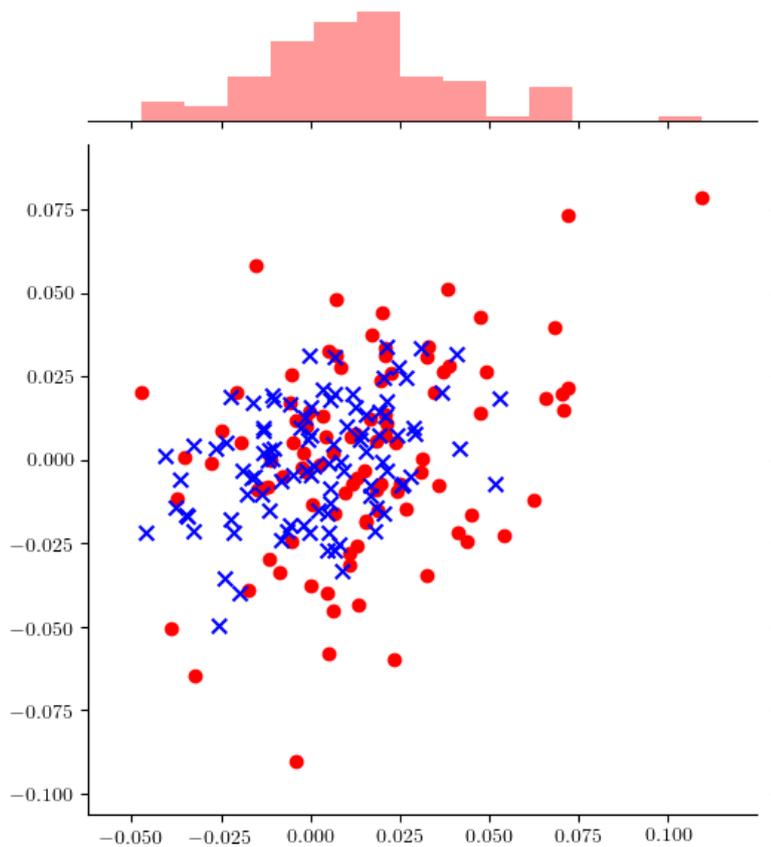


Figura 5.12: Distribuição conjunta real e gerada pela ER-GAN de dois ativos do portfólio para o dataset 100 Ativos do S&P 500. Pode-se observar que a distribuição da ER-GAN gera amostras bem dispersas e com uma forte sobreposição com a distribuição original, conforme o esperado.

que apresentou o maior valor de razão de sharpe foi 0.005.

O melhor tamanho de janela manteve-se sendo de 20 amostras, valor usado durante o treinamento com o dataset de 10 ativos, como pode ser visto na Tabela 5.5

Tabela 5.10: Comparação entre taxas de aprendizado para o dataset 100 Ativos do S&P 500.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	Razão de Sharpe ($\times 10^{-2}$)
0,0001	7,582 \pm 0.180	0,890 \pm 0,005	7,638 \pm 0,205
0,0005	7,659 \pm 0.222	0,915 \pm 0,016	7,616 \pm 0,275
0,001	7,204 \pm 0.427	0,897 \pm 0,012	7,239 \pm 0,474
0,005	6,840 \pm 0.529	0,950 \pm 0,016	6,447 \pm 0,559
0,01	6,562 \pm 0.319	0,907 \pm 0,014	6,440 \pm 0,368

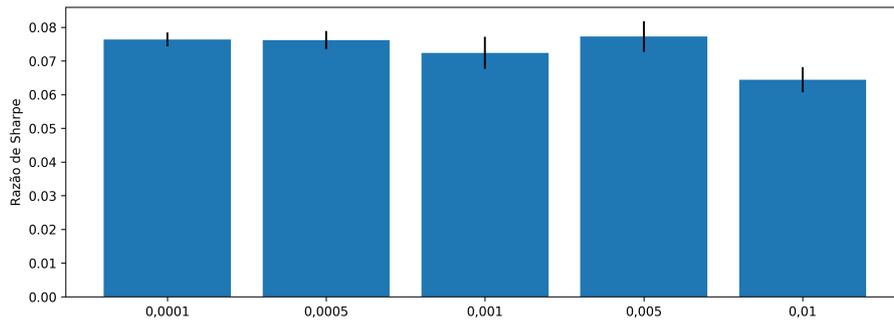


Figura 5.13: Comparação entre taxas de aprendizado para o dataset 100 Ativos do S&P 500.

e na Figura 5.7.

Tabela 5.11: Comparação entre tamanhos de janela para o dataset 100 Ativos do S&P 500.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
20	7,145 \pm 0.399	0,943 \pm 0,017	6,925 \pm 0,452
50	6,229 \pm 0.394	0,899 \pm 0,010	6,082 \pm 0,441
100	7,359 \pm 0.417	0,936 \pm 0,016	7,119 \pm 0,442

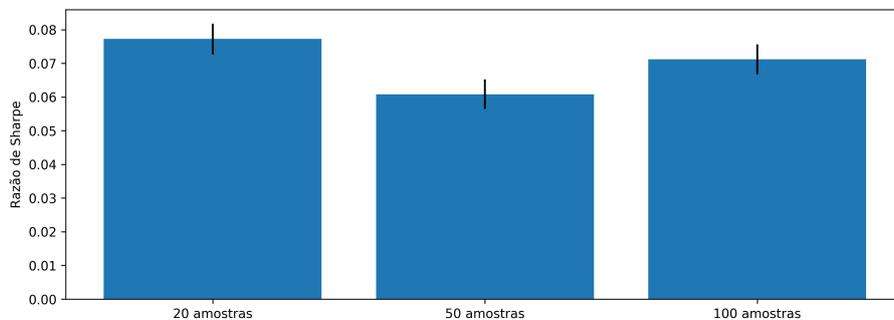


Figura 5.14: Comparação entre tamanhos de janela para o dataset 100 Ativos do S&P 500.

A seguir comparou-se o tipo de treinamento utilizado pela GAN, mais uma vez o treinamento padrão foi o que apresentou o melhor resultado, como pode ser visto na Tabela 5.6 e na Figura 5.8.

Tabela 5.12: Comparação entre métodos de treinamento de GAN para o dataset 100 Ativos do S&P 500.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
GAN	$8,325 \pm 0,508$	$0,942 \pm 0,020$	$7,950 \pm 0,464$
WGAN	$7,490 \pm 0,222$	$0,903 \pm 0,007$	$7,451 \pm 0,262$
WGAN-GP	$7,570 \pm 1,497$	$0,937 \pm 0,037$	$7,179 \pm 1,311$

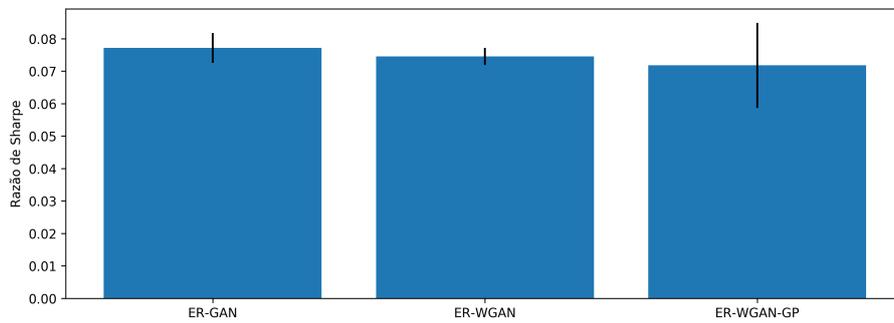


Figura 5.15: Comparação entre métodos de treinamento de GAN para o dataset 100 Ativos do S&P 500.

Por último avaliou-se o uso de discriminação de mini-lotes e normalização por mini-lotes. Como o dataset de 100 ativos observou-se que o modelo utilizando discriminação por minilotes conseguiu convergir em apenas um dos 50 cenários avaliados de forma que optou-se por desconsiderar o método na comparação. Levando em consideração apenas a comparação entre o modelo ER-GAN simples e o modelo ER-GAN utilizando normalização por minilotes observou-se que o modelo simples apresentou resultado superior, tanto em retorno acumulado quando do ponto de vista do Razão de Sharpe, como pode ser visto na Tabela 5.7 e na Figura 5.9. Mais uma vês constatou-se que o modelo ER-GAN não sofreu de colapso de modo, de forma que as técnicas utilizadas na literatura não se mostraram eficazes.

Tabela 5.13: Comparação entre métodos para a melhoria da diversidade das amostras com o dataset 100 Ativos do S&P 500.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
GAN	$8,325 \pm 0,508$	$0,942 \pm 0,020$	$7,950 \pm 0,464$
B Norm,	$7,802 \pm 0,755$	$1,105 \pm 0,043$	$6,431 \pm 0,625$

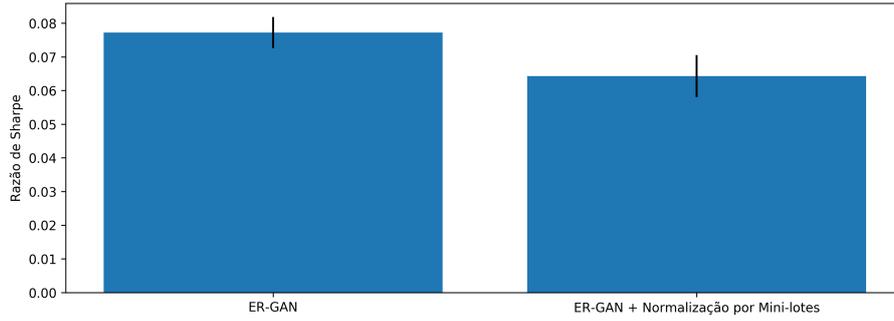


Figura 5.16: Comparação do retorno acumulado entre métodos para a melhoria da diversidade das amostras com o dataset 100 Ativos do S&P 500.

Para este dataset não foi necessário treinar o modelo combinando os melhores parâmetros encontrados, uma vez que os parâmetros que apresentaram os melhores resultados já eram aqueles que foram usados no experimento para determinar a topologia da rede. O valor médio da Razão de Sharpe desse modelo nas amostras de validação foi de $(7.100 \pm 0,227) \times 10^{-2}$. Em seguida, foi selecionado o modelo que apresentou o maior retorno médio entre os 50 treinados. Este foi usado para comparar o modelo com os benchmarks usando o conjunto de testes. O resultado da média, variância e Razão de Sharpe pode ser visto na Tabela 5.8. O modelo ER-GAN apresentou um valor médio de retorno mais alto e uma Razão de Sharpe maior que todos os outros benchmarks. Seu retorno médio foi 34,55% maior que o benchmark com maior retorno e sua Razão de Sharpe foi 27,26% maior que o melhor benchmark. A diferença foi maior do que a obtida usando o dataset de 10 ativos. Isso indica que as estimativas da ER-GAN para o retorno médio e para a matriz de covariâncias apresentam menor sensibilidade ao aumento do tamanho do portfólio do que os benchmarks.

Tabela 5.14: Comparação entre o modelo ER-GAN e benchmarks para o dataset 100 Ativos do S&P 500. Seu retorno médio foi 34,55% maior que o benchmark com maior retorno e sua Razão de Sharpe foi 27,26% maior que o melhor benchmark.

	Média ($\times 10^{-4}$)	Variância ($\times 10^{-2}$)	SR ($\times 10^{-2}$)
MV	0,165	0,120	1,372
EW	0,123	0,105	1,175
SFS	0,135	0,121	1,117
CCS	0,115	0,125	0,922
CVS	0,158	0,120	1,313
OAS	0,159	0,120	1,326
ER-GAN	0,222	0,127	1,746

A Figura 5.17 mostra o retorno cumulativo de todos os modelos. Fica evidente

que o retorno cumulativo do ER-GAN permanece superior aos demais durante praticamente todo o período analisado, sendo menos suscetíveis a grandes variações de retornos do que os benchmarks.

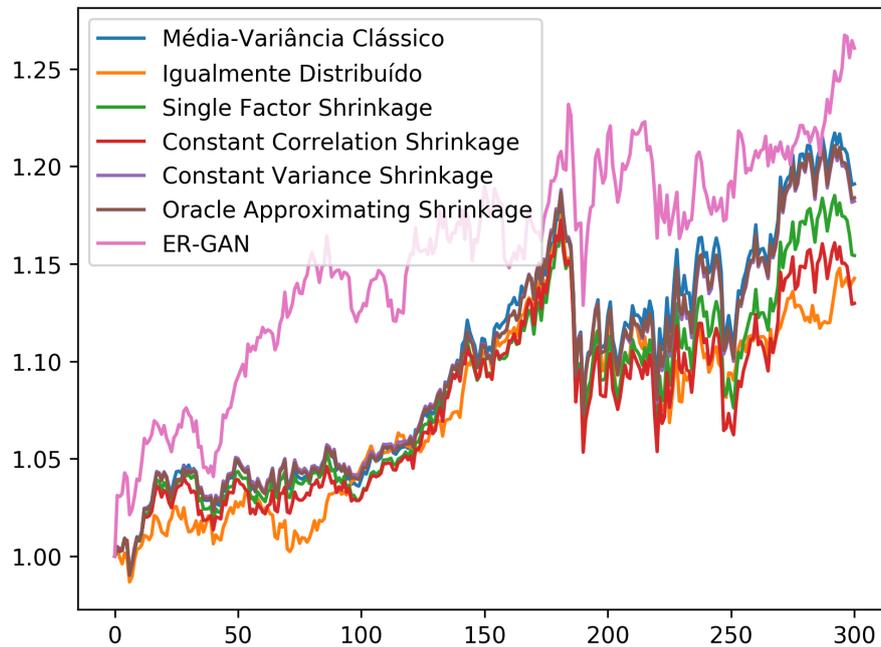


Figura 5.17: Comparação do retorno acumulado entre a melhor configuração de ER-GAN e os benchmarks para o dataset 100 Ativos do S&P 500. Fica evidente que o retorno cumulativo do ER-GAN permanece superior aos demais durante praticamente todo o período analisado, sendo menos suscetíveis a grandes variações de retornos do que os benchmarks.

Capítulo 6

Conclusões

Neste trabalho, foi proposto um modelo chamado ER-GAN, aplicando pela primeira vez GANs recorrentes ao problema de seleção de portfólio. Também foi proposta uma nova técnica para o treinamento de modelos GAN aplicados a dados sequenciais. A técnica usa uma rede recorrente para modelar a dependência temporal dos dados além da rede a ser treinada com o erro do adversário. Este procedimento garante a diversidade das amostras geradas e a convergência do treinamento.

Os resultados obtidos mostram que o modelo ER-GAN é capaz de modelar a distribuição estatística conjunta de ativos em um portfólio. O modelo foi capaz de apresentar resultados de Razão de Sharpe superiores a todos os benchmarks analisados, tanto ao ser aplicado a datasets pequenos (10 ativos) quanto a datasets grandes (100 ativos). Seu retorno médio foi 34,55% superior ao benchmark com retornos mais altos e sua Razão de Sharpe 27,26% superior ao melhor benchmark ao ser aplicado para o dataset de 100 ativos, demonstrando ser menos sensível a variações no tamanho do portfólio do que os benchmarks.

Possíveis contribuições futuras incluem: aplicar o modelo ER-GAN à seleção de portfólio usando diferentes métricas de risco, como VaR e CVaR; avaliação de novas técnicas de treinamento GAN aplicadas ao problema de seleção de portfólio; aplicação de modelo para seleção de portfólio de vários períodos usando ER-GAN.

Referências Bibliográficas

- [1] MARKOWITZ, H. “Portfolio selection”, *The Journal of Finance*, v. 7, n. 1, pp. 77–91, 1952.
- [2] MARKOWITZ, H. *Portfolio selection: efficient diversification of investments*. Cowles foundation for research in economics at Yale university. Monograph. New York, John Wiley & Sons, 1959.
- [3] DEMIGUEL, V., GARLAPPI, L., UPPAL, R. “Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?” *The review of Financial studies*, v. 22, n. 5, pp. 1915–1953, 2007.
- [4] STĂRICĂ, C., GRANGER, C. “Nonstationarities in stock returns”, *Review of economics and statistics*, v. 87, n. 3, pp. 503–522, 2005.
- [5] MIKOSCH, T., STĂRICĂ, C. “Nonstationarities in financial time series, the long-range dependence, and the IGARCH effects”, *Review of Economics and Statistics*, v. 86, n. 1, pp. 378–390, 2004.
- [6] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- [7] ZHOU, X., PAN, Z., HU, G., et al. “Stock market prediction on high-frequency data using generative adversarial nets”, *Mathematical Problems in Engineering*, v. 2018, 2018.
- [8] ELTON, E. J., GRUBER, M. J. “Modern portfolio theory, 1950 to date”, *Journal of Banking & Finance*, v. 21, n. 11-12, pp. 1743–1759, 1997.
- [9] BODIE, Z., KANE, A., MARCUS, A. *Investments*. Cowles foundation for research in economics at Yale university. Monograph. 10 ed. New York, McGraw-Hill Education, 2013.
- [10] SHARPE, W. F. “A simplified model for portfolio analysis”, *Management science*, v. 9, n. 2, pp. 277–293, 1963.

- [11] SHARPE, W. F. “Capital asset prices: A theory of market equilibrium under conditions of risk”, *The journal of finance*, v. 19, n. 3, pp. 425–442, 1964.
- [12] ELTON, E. J., GRUBER, M. J. “Estimating the dependence structure of share prices—implications for portfolio selection”, *The Journal of Finance*, v. 28, n. 5, pp. 1203–1232, 1973.
- [13] KING, B. F. “Market and industry factors in stock price behavior”, *the Journal of Business*, v. 39, n. 1, pp. 139–190, 1966.
- [14] COHEN, K. J., POGUE, J. A. “An empirical evaluation of alternative portfolio-selection models”, *The Journal of Business*, v. 40, n. 2, pp. 166–193, 1967.
- [15] FAMA, E. F., FRENCH, K. R. “The cross-section of expected stock returns”, *the Journal of Finance*, v. 47, n. 2, pp. 427–465, 1992.
- [16] FRENCH, K. R. “Kenneth R. French - Data Library”. Disponível em: <http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.htm>. Acessado em: 2018-12-04.
- [17] FAMA, E. F., FRENCH, K. R. “A five-factor asset pricing model”, *Journal of Financial Economics*, v. 116, n. 1, pp. 1–22, apr 2015. doi: 10.1016/j.jfineco.2014.10.010. Disponível em: <<https://doi.org/10.1016/j.jfineco.2014.10.010>>.
- [18] FAMA, E. F., FRENCH, K. R. “Choosing factors”, *Journal of Financial Economics*, v. 128, n. 2, pp. 234–252, 2018.
- [19] PELGER, M. “Large-dimensional factor modeling based on high-frequency observations”, *Journal of Econometrics*, 2018.
- [20] AÏT-SAHALIA, Y., XIU, D. “Using principal component analysis to estimate a high dimensional factor model with high-frequency data”, *Journal of Econometrics*, v. 201, n. 2, pp. 384–399, 2017.
- [21] HARVEY, C. R., LIU, Y., ZHU, H. “... and the cross-section of expected returns”, *The Review of Financial Studies*, v. 29, n. 1, pp. 5–68, 2016.
- [22] STEIN, C. *Inadmissibility of the usual estimator for the mean of a multivariate normal distribution*. Relatório técnico, Stanford University Stanford United States, 1956.
- [23] FROST, P. A., SAVARINO, J. E. “An empirical Bayes approach to efficient portfolio selection”, *Journal of Financial and Quantitative Analysis*, v. 21, n. 3, pp. 293–305, 1986.

- [24] JORION, P. “Bayes-Stein estimation for portfolio analysis”, *Journal of Financial and Quantitative Analysis*, v. 21, n. 3, pp. 279–292, 1986.
- [25] LEDOIT, O., WOLF, M. “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection”, *Journal of empirical finance*, v. 10, n. 5, pp. 603–621, 2003.
- [26] LEDOIT, O., WOLF, M. “Honey, I shrunk the sample covariance matrix”, *The Journal of Portfolio Management*, v. 30, n. 4, pp. 110–119, 2004.
- [27] LEDOIT, O., WOLF, M. “A well-conditioned estimator for large-dimensional covariance matrices”, *Journal of multivariate analysis*, v. 88, n. 2, pp. 365–411, 2004.
- [28] CHEN, Y., WIESEL, A., ELDAR, Y. C., et al. “Shrinkage algorithms for MMSE covariance estimation”, *IEEE Transactions on Signal Processing*, v. 58, n. 10, pp. 5016–5029, 2010.
- [29] RUTECKI, P. A. “Neuronal excitability: voltage-dependent currents and synaptic transmission.” *Journal of clinical neurophysiology: official publication of the American Electroencephalographic Society*, v. 9, n. 2, pp. 195–211, 1992.
- [30] HERCULANO-HOUZEL, S. “The human brain in numbers: a linearly scaled-up primate brain”, *Frontiers in human neuroscience*, v. 3, pp. 31, 2009.
- [31] MCCULLOCH, W. S., PITTS, W. “A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysics*, v. 5, n. 4, pp. 115–133, 1943.
- [32] ROSENBLATT, F. “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, v. 65, n. 6, pp. 386, 1958.
- [33] WERBOS, P. “Beyond Regression:” New Tools for Prediction and Analysis in the Behavioral Sciences”, *Ph. D. dissertation, Harvard University*, 1974.
- [34] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J., et al. “Learning representations by back-propagating errors”, *Cognitive modeling*, v. 5, n. 3, pp. 1, 1988.
- [35] CYBENKO, G. “Approximation by superpositions of a sigmoidal function”, *Mathematics of control, signals and systems*, v. 2, n. 4, pp. 303–314, 1989.

- [36] KELLEY, H. J. “Gradient theory of optimal flight paths”, *Ars Journal*, v. 30, n. 10, pp. 947–954, 1960.
- [37] MIKOLOV, T., KARAFIÁT, M., BURGET, L., et al. “Recurrent neural network based language model”. In: *Eleventh annual conference of the international speech communication association*, 2010.
- [38] GRAVES, A., MOHAMED, A.-R., HINTON, G. “Speech recognition with deep recurrent neural networks”. In: *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. IEEE, 2013.
- [39] LAI, S., XU, L., LIU, K., et al. “Recurrent convolutional neural networks for text classification”. In: *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
- [40] LIU, P., QIU, X., HUANG, X. “Recurrent neural network for text classification with multi-task learning”, *arXiv preprint arXiv:1605.05101*, 2016.
- [41] VENUGOPALAN, S., XU, H., DONAHUE, J., et al. “Translating videos to natural language using deep recurrent neural networks”, *arXiv preprint arXiv:1412.4729*, 2014.
- [42] YU, H., WANG, J., HUANG, Z., et al. “Video paragraph captioning using hierarchical recurrent neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4584–4593, 2016.
- [43] SAK, H., SENIOR, A., BEAUFAYS, F. “Long short-term memory recurrent neural network architectures for large scale acoustic modeling”. In: *Fifteenth annual conference of the international speech communication association*, 2014.
- [44] WANG, J.-H., LEU, J.-Y. “Stock market trend prediction using ARIMA-based neural networks”. In: *Proceedings of International Conference on Neural Networks (ICNN'96)*, v. 4, pp. 2160–2165. IEEE, 1996.
- [45] SAAD, E. W., PROKHOROV, D. V., WUNSCH, D. C. “Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks”, *IEEE Transactions on neural networks*, v. 9, n. 6, pp. 1456–1470, 1998.
- [46] HSIEH, T.-J., HSIAO, H.-F., YEH, W.-C. “Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm”, *Applied soft computing*, v. 11, n. 2, pp. 2510–2525, 2011.

- [47] RATHER, A. M., AGARWAL, A., SASTRY, V. “Recurrent neural network and a hybrid model for prediction of stock returns”, *Expert Systems with Applications*, v. 42, n. 6, pp. 3234–3241, 2015.
- [48] BAO, W., YUE, J., RAO, Y. “A deep learning framework for financial time series using stacked autoencoders and long-short term memory”, *PloS one*, v. 12, n. 7, pp. e0180944, 2017.
- [49] ELMAN, J. L. “Finding structure in time”, *Cognitive science*, v. 14, n. 2, pp. 179–211, 1990.
- [50] JORDAN, M. “Attractor dynamics and parallelism in a connectionist sequential machine”. In: *Proc. of the Eighth Annual Conference of the Cognitive Science Society (Erlbaum, Hillsdale, NJ), 1986*, 1986.
- [51] BENGIO, Y., SIMARD, P., FRASCONI, P., et al. “Learning long-term dependencies with gradient descent is difficult”, *IEEE transactions on neural networks*, v. 5, n. 2, pp. 157–166, 1994.
- [52] HOCHREITER, S. “Untersuchungen zu dynamischen neuronalen Netzen”, *Diploma, Technische Universität München*, v. 91, n. 1, 1991.
- [53] DOYA, K. “Bifurcations of recurrent neural networks in gradient descent learning”, *IEEE Transactions on neural networks*, v. 1, n. 75, pp. 164, 1993.
- [54] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. “Deep learning”. p. 404, MIT press, 2016.
- [55] HOCHREITER, S., SCHMIDHUBER, J. “Long short-term memory”, *Neural computation*, v. 9, n. 8, pp. 1735–1780, 1997.
- [56] KAMIJO, K.-I., TANIGAWA, T. “Stock price pattern recognition-a recurrent neural network approach”. In: *1990 IJCNN International Joint Conference on Neural Networks*, pp. 215–221. IEEE, 1990.
- [57] CHEN, K., ZHOU, Y., DAI, F. “A LSTM-based method for stock returns prediction: A case study of China stock market”. In: *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2823–2824. IEEE, 2015.
- [58] HUYNH, H. D., DANG, L. M., DUONG, D. “A new model for stock price movements prediction using deep neural network”. In: *Proceedings of the Eighth International Symposium on Information and Communication Technology*, pp. 57–62. ACM, 2017.

- [59] DING, X., ZHANG, Y., LIU, T., et al. “Using structured events to predict stock price movement: An empirical investigation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1415–1425, 2014.
- [60] PENG, Y., JIANG, H. “Leverage financial news to predict stock price movements using word embeddings and deep neural networks”, *arXiv preprint arXiv:1506.07220*, 2015.
- [61] KIM, H. Y., WON, C. H. “Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models”, *Expert Systems with Applications*, v. 103, pp. 25–37, 2018.
- [62] RADFORD, A., METZ, L., CHINTALA, S. “Unsupervised representation learning with deep convolutional generative adversarial networks”, *arXiv preprint arXiv:1511.06434*, 2015.
- [63] ZHU, J.-Y., KRÄHENBÜHL, P., SHECHTMAN, E., et al. “Generative visual manipulation on the natural image manifold”. In: *European Conference on Computer Vision*, pp. 597–613. Springer, 2016.
- [64] WU, J., ZHANG, C., XUE, T., et al. “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling”. In: *Advances in neural information processing systems*, pp. 82–90, 2016.
- [65] LEDIG, C., THEIS, L., HUSZÁR, F., et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.
- [66] SALIMANS, T., GOODFELLOW, I., ZAREMBA, W., et al. “Improved techniques for training gans”. In: *Advances in neural information processing systems*, pp. 2234–2242, 2016.
- [67] ARJOVSKY, M., CHINTALA, S., BOTTOU, L. “Wasserstein gan”, *arXiv preprint arXiv:1701.07875*, 2017.
- [68] HÖRMANDER, F. H. N. H. L., TOTARO, N. S. B., WALDSCHMIDT, A. V. M. “Grundlehren der mathematischen Wissenschaften 332”, 2006.
- [69] GULRAJANI, I., AHMED, F., ARJOVSKY, M., et al. “Improved training of wasserstein gans”. In: *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

- [70] IOFFE, S., SZEGEDY, C. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *arXiv preprint arXiv:1502.03167*, 2015.
- [71] SZEGEDY, C., VANHOUCKE, V., IOFFE, S., et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [72] LOTTER, W., KREIMAN, G., COX, D. “Unsupervised learning of visual structure using predictive generative networks”, *arXiv preprint arXiv:1511.06380*, 2015.
- [73] YU, L., ZHANG, W., WANG, J., et al. “Seqgan: Sequence generative adversarial nets with policy gradient”. In: *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [74] LAMB, A. M., GOYAL, A. G. A. P., ZHANG, Y., et al. “Professor forcing: A new algorithm for training recurrent networks”. In: *Advances In Neural Information Processing Systems*, pp. 4601–4609, 2016.
- [75] ESTEBAN, C., HYLAND, S. L., RÄTSCHE, G. “Real-valued (medical) time series generation with recurrent conditional gans”, *arXiv preprint arXiv:1706.02633*, 2017.
- [76] CHEN, Y., WANG, Y., KIRSCHEN, D., et al. “Model-free renewable scenario generation using generative adversarial networks”, *IEEE Transactions on Power Systems*, v. 33, n. 3, pp. 3265–3275, 2018.
- [77] GUPTA, A., JOHNSON, J., FEI-FEI, L., et al. “Social gan: Socially acceptable trajectories with generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.
- [78] KINGMA, D. P., BA, J. “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.
- [79] BACHELIER, L. “Théorie de la spéculation”. In: *Annales scientifiques de l’École normale supérieure*, v. 17, pp. 21–86, 1900.
- [80] BLACK, F., SCHOLES, M. “The pricing of options and corporate liabilities”, *Journal of political economy*, v. 81, n. 3, pp. 637–654, 1973.
- [81] MIRZA, M., OSINDERO, S. “Conditional generative adversarial nets”, *arXiv preprint arXiv:1411.1784*, 2014.

- [82] MAVERICK, J. B. “Most Commonly-Used Periods in Creating Moving Average (MA) Lines”. Disponível em: <https://www.investopedia.com/ask/answers/122414/what-are-most-common-periods-used-creating-moving-average-ma-lines.asp>. Accessed on: 2019-11-02.
- [83] 2019. Disponível em: <https://www.alphavantage.co>.

Apêndice A

Evolução do Treinamento com Dados Sintéticos

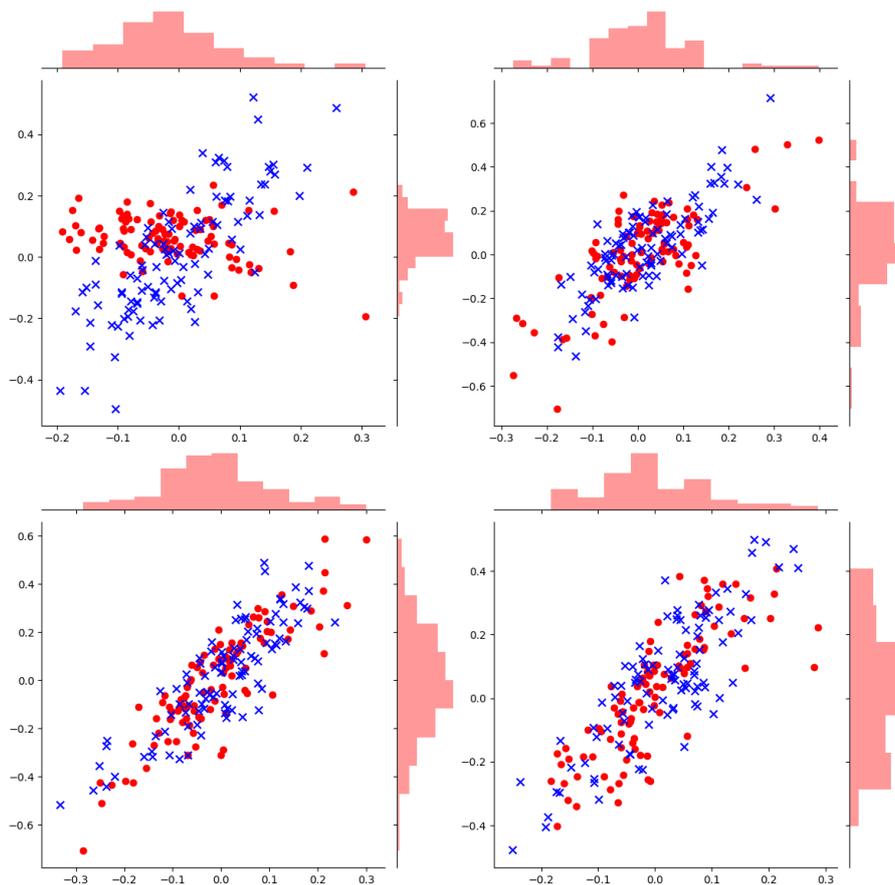


Figura A.1: Evolução da qualidade das amostras geradas pela GAN para dados sintéticos nas épocas 0 (cima-esquerda), 50 (cima-direita), 150 (baixo-esquerda) e 200.

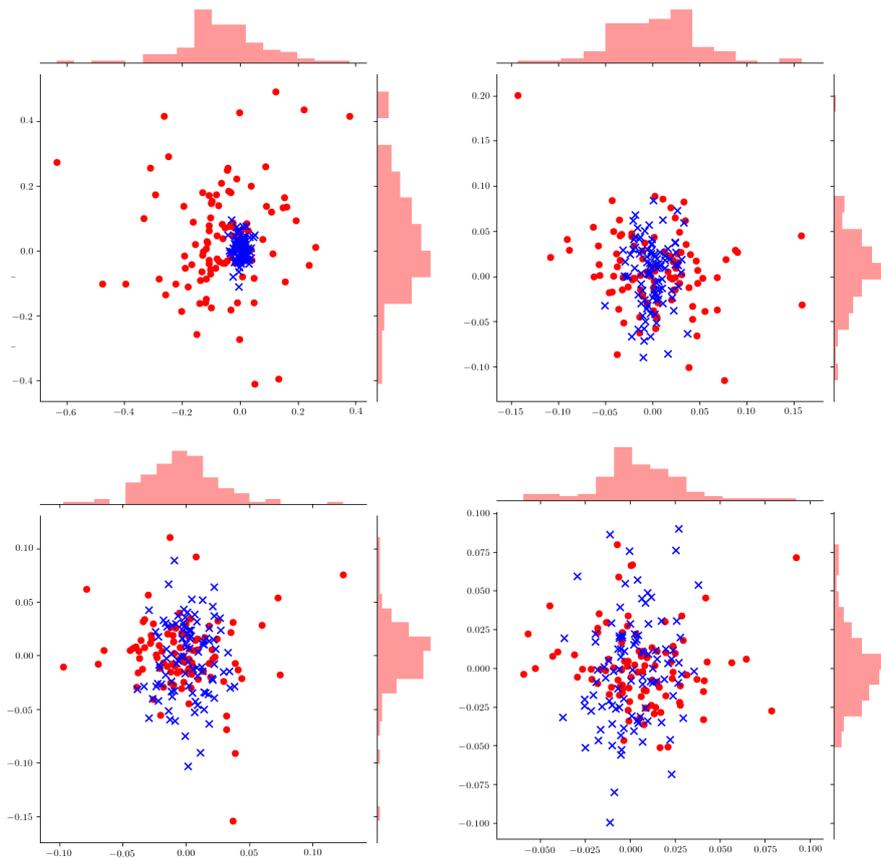


Figura A.2: Evolução da qualidade das amostras geradas pela ER-GAN para o dataset Um Ativo por Indústria nas épocas 0 (cima-esquerda), 50 (cima-direita), 150 (baixo-esquerda) e 200.

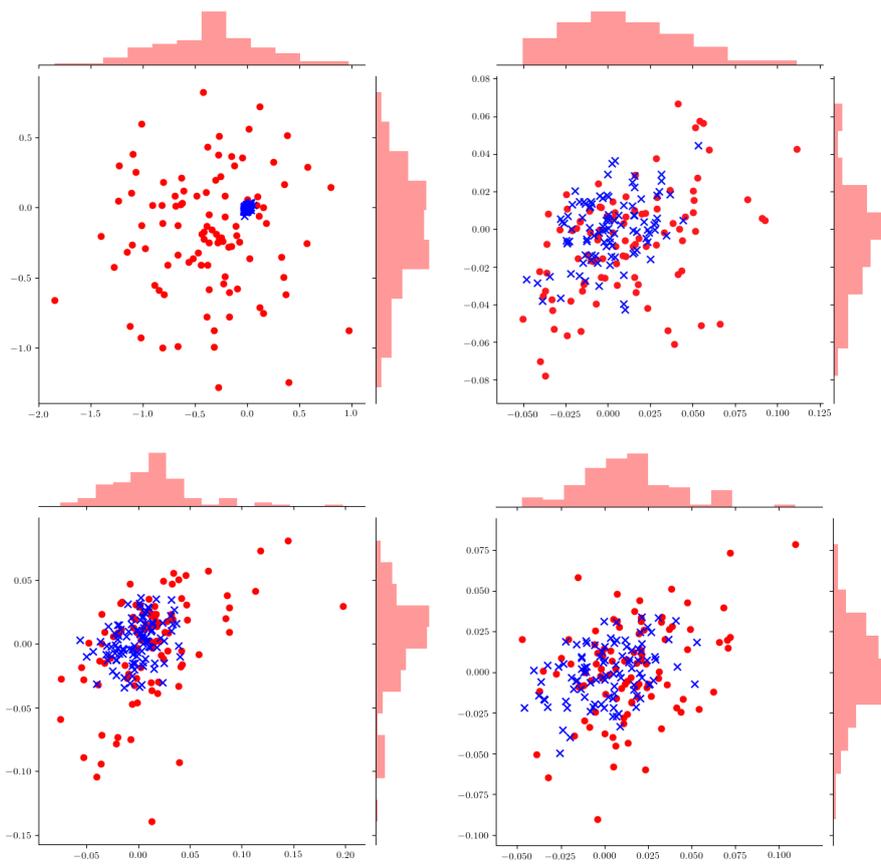


Figura A.3: Evolução da qualidade das amostras geradas pela ER-GAN para o dataset 100 Ativos do S&P 500 nas épocas 0 (cima-esquerda), 100 (cima-direita), 200 (baixo-esquerda) e 300.

Apêndice B

Ativos Presentes no Dataset Um Ativo por Indústria

Tabela B.1: Comparação entre métodos para a melhoria da diversidade das amostras com o dataset Um Ativo por Indústria.

Ticker	Setor	Empresa
AMT	Imobiliário	American Tower Corp.
AMZN	Consumo de Luxo	Amazon.com, Inc.
BA	Indústria	The Boeing Company
BRK-B	Financeiro	Berkshire Hathaway Inc. New
ECL	Matérias Primas	Ecolab Inc
MSFT	Tecnologia da Informação	Microsoft Corp.
NEE	Utilidades Públicas	Nextera Energy, Inc.
UNH	Saúde	Unitedhealth Group Incorporated
VZ	Telecomunicações	Verizon Communications Inc.
WMT	Consumo de primeira necessidade	Walmart Inc.
XOM	Energia	Exxon Mobil Corporation

Apêndice C

Ativos Presentes no Dataset 100 Ativos do S&P 500

Tabela C.1: Ações do dataset 100 Ativos do S&P 500

Ticker	Empresa
ADM	Archer-Daniels-Midland Company
AON	Aon plc
CB	Chubb Limited
CPB	Campbell Soup Company
DVN	Devon Energy Corporation
ETFC	E*TRADE Financial Corporation
HES	Hess Corporation
IRM	Iron Mountain Incorporated
KR	The Kroger Co.
LRCX	Lam Research Corporation
MDT	Medtronic plc
NI	NiSource Inc.
PFE	Pfizer Inc.
PSA	Public Storage
TGT	Target Corporation
WHR	Whirlpool Corporation
ADP	Automatic Data Processing, Inc.
APC	Anadarko Petroleum Corporation
CELG	Celgene Corporation
CPRT	Copart, Inc.
EA	Electronic Arts Inc.
ETN	Eaton Corporation plc

Continua na próxima página

Tabela C.1 – continuação

Ticker	Empresa
GE	General Electric Company
HRB	H&R Block, Inc.
IT	Gartner, Inc.
KSU	Kansas City Southern
MAS	Masco Corporation
MLM	Martin Marietta Materials, Inc.
NTRS	Northern Trust Corporation
PG	The Procter & Gamble Company
PXD	Pioneer Natural Resources Company
TSCO	Tractor Supply Company
WMT	Walmart Inc.
AEE	Ameren Corporation
ARE	Alexandria Real Estate Equities, Inc.
CMCSA	Comcast Corporation
DHI	D.R. Horton, Inc.
EL	The Estée Lauder Companies Inc.
EXC	Exelon Corporation
GLW	Corning Incorporated
HRL	Hormel Foods Corporation
JBHT	J.B. Hunt Transport Services, Inc.
LB	L Brands, Inc.
MAT	Mattel, Inc.
MMM	3M Company
NUE	Nucor Corporation
PHM	PulteGroup, Inc.
ROL	Rollins, Inc.
TSS	Total System Services, Inc.
WY	Weyerhaeuser Company
AES	The AES Corporation
AVY	Avery Dennison Corporation
CMI	Cummins Inc.
DISH	DISH Network Corporation
EMN	Eastman Chemical Company
FCX	Freeport-McMoRan Inc.
GWW	W.W. Grainger, Inc.
HRS	L3Harris Technologies, Inc.

Continua na próxima página

Tabela C.1 – continuação

Ticker	Empresa
K	Kellogg Company
LEG	Leggett & Platt, Incorporated
MCK	McKesson Corporation
MO	Altria Group, Inc.
ORCL	Oracle Corporation
PNR	Pentair plc
ROP	Roper Technologies, Inc.
UNM	Unum Group
XRAY	DENTSPLY SIRONA Inc.
AJG	Arthur J. Gallagher & Co.
BBY	Best Buy Co., Inc.
CMS	CMS Energy Corporation
DRE	Duke Realty Corporation
EOG	EOG Resources, Inc.
FE	FirstEnergy Corp.
HCP	HCP, Inc.
HSY	The Hershey Company
KMB	Kimberly-Clark Corporation
LNC	Lincoln National Corporation
MCO	Moody's Corporation
MYL	Mylan N.V.
PCG	PG&E Corporation
PNW	Pinnacle West Capital Corporation
RTN	Raytheon Company
VAR	Varian Medical Systems, Inc.
XRX	Xerox Corporation
ALL	The Allstate Corporation
BXP	Boston Properties, Inc.
COST	Costco Wholesale Corporation
DRI	Darden Restaurants, Inc.
ESS	Essex Property Trust, Inc.
FITB	Fifth Third Bancorp
HD	The Home Depot, Inc.
HUM	Humana Inc.
KMX	CarMax, Inc.
LOW	Lowe's Companies, Inc.

Continua na próxima página

Tabela C.1 – continuação

Ticker	Empresa
M	Macy's, Inc.
NBL	Noble Energy, Inc.
PEG	Public Service Enterprise Group Incorporated
PRGO	Perrigo Company plc
STZ	Constellation Brands, Inc.
WDC	Western Digital Corporation