

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
INSTITUTO DE FÍSICA

ESTUDO DE SÉRIES TEMPORAIS E SUAS APLICAÇÕES
EM ECONOFÍSICA USANDO REDES NEURAIS
ARTIFICIAIS

DIEGO RENATO PESSOA

Uberlândia
2025

DIEGO RENATO PESSOA

**ESTUDO DE SÉRIES TEMPORAIS E SUAS APLICAÇÕES EM
ECONOFÍSICA USANDO REDES NEURAIS ARTIFICIAIS**

Trabalho de Conclusão de Curso apresentado
ao Instituto de Física da Universidade Federal
de Uberlândia, como requisito parcial para
obtenção do título de bacharel em Física Médica.

Orientador:

Prof. Dr. João Carlos de Oliveira Guerra .

**Uberlândia
2025**

DIEGO RENATO PESSOA

**ESTUDO DE SÉRIES TEMPORAIS E SUAS APLICAÇÕES EM
ECONOFÍSICA USANDO REDES NEURAIS ARTIFICIAIS**

Trabalho de Conclusão de Curso apresentado ao Instituto de Física da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de bacharel em Física Médica.

Uberlândia, 22 de Setembro de 2025.

Banca Examinadora:

Prof. Dr. João Carlos de Oliveira Guerra
Orientador (INFIS - UFU)

Prof. Dr. Jean Venato Santos
Membro Interno (IME - UFU)

Prof. Dr. Fabricio Macedo de Souza
Membro Interno (INFIS - UFU)

FICHA CATALOGRÁFICA

Para o Sábio Autor das leis da natureza, que nos permite sondar os mistérios do universo com a razão que nos foi dada e o encanto que nos rodeia. Antes de procurá-Lo nos enigmas, olhemos ao redor para ver como Ele sorri nas criaturas.

(Jean Piaget, Fr R.U. Ameixeira)

AGRADECIMENTOS

Agradeço primeiro a Deus e pela intercessão da Virgem Maria, a Senhora de Aparecida, que concebeu a mim a graça do tão sonhado acesso a Universidade Pública gratuita e de qualidade e principalmente, por me dar força para trilhar este sonho, pois, não é segredo que a permanência em uma cidade diferente, longe da família e com poucos recursos não é fácil;

Aos meus pais, Sonia e Claudio, que fizeram o possível e o impossível em meio a tantas dificuldades para que eu ingressasse em uma universidade, espero que eles saibam que o meu sucesso nada mais é que fruto do esforço, das orações e renúncias diária deles, por isso, muito obrigado! Amo-vos profundamente;

A minha noiva, Mariana, que é meu conforto, minha companheira, minha melhor amiga e tem sido meu apoio em todos os momentos até aqui, tenham sido eles dolorosos ou alegres. Não foram poucos os momentos que me espelhei e perseverei por sua causa. Te agradeço por tudo! Te amo, meu bem;

Aos meus melhores amigos de infância, André, Miriã e Isabela que sempre se esforçaram, incentivaram e foram o conforto necessário mesmo a distância;

Aos meus professores do Ensino Médio, em especial Iranice, Eucimara, Maria Helena, Paula, Carolina, Edimilson e Silvio, que foram minha inspiração para seguir o caminho da educação e da ciência, apesar de tantas dificuldades. Vou levá-los sempre em meu coração;

Ao meu orientador, professor e amigo, João Carlos, que me auxiliou durante toda esta fase com paciência, conselhos e zelo. Estendo minha gratidão aos professores e técnicos do INFIS, em especial Lúcio, Ana, Mauricio e Akira, que direta ou indiretamente contribuíram para minha formação. Agradeço igualmente aos professores Jean Venato Santos e Fabricio Macedo de Souza, membros da banca, pela honra de sua presença e avaliação neste dia tão importante.

E por fim, mas, não menos importante, a todos aqueles que me acolheram nesta nova cidade e que hoje tenho o prazer de chamar de amigos. A estes, muito obrigado por estarem presentes em tantos momentos importantes da minha vida. Vou levá-los sempre em meu coração! Em especial: Priscila, Kauêne, Ana Luiza, Maria das Graças, Adrielly e Isabelle, muito obrigado por tudo!

*“A principal meta da educação é criar homens que
sejam capazes de fazer coisas novas, não
simplesmente repetir o que outras gerações já fizeram.*

*Homens que sejam criadores, inventores,
descobridores. A segunda meta da educação é formar
mentes que estejam em condições de criticar, verificar
e não aceitar tudo que a elas se propõe.”*

Jean Piaget

RESUMO

Este trabalho insere-se no campo da Econofísica, e teve como objetivo criar e avaliar a eficácia de uma Rede Neural Artificial (RNA) com arquitetura Long Short-Term Memory (LSTM) na previsão de séries temporais de preços de ações do mercado de capitais brasileiro. Para isso, foram coletados dados históricos diários dos ativos Petrobras (PETR4.SA), Vale (VALE3.SA) e Itaú (ITUB4.SA) no período de 2005 a 2025. A metodologia envolveu a utilização de modelos computacionais avançados para analisar sistemas financeiros complexos e incluiu como parte do pré-processamento dos dados a normalização utilizando a função(MinMaxScaler) e a estruturação em janelas temporais de 60 dias para alimentar um modelo LSTM profundo, que possui quatro camadas e regularizado com a técnica conhecida como Dropout (taxa de 0.2) para prevenir o superajustamento (overfitting). Os resultados aplicados no conjunto de teste revelaram um desempenho variado: alcançou-se uma alta precisão para o ativo do setor bancário (Itaú), uma boa capacidade de previsão de tendência para o setor de petróleo (Petrobras) e uma falha quantitativamente significativa para o ativo do setor de mineração (Vale), evidenciada por um erro significativo e um "cluster"no final da época de treinamento somente para o setor da Vale. Conclui-se que, apesar das redes LSTM apresentarem um grande potencial para a modelagem de séries temporais financeiras, sua eficácia depende fortemente das características particulares e da estabilidade de cada ativo, o que reforça a ideia de que o mercado financeiro é um sistema complexo e não universal.

Palavras-chave: Séries temporais, Econofísica, Redes Neurais Artificiais.

Abstract

This work falls within the field of Econophysics and aimed to create and evaluate the effectiveness of an Artificial Neural Network (ANN) with a Long Short-Term Memory (LSTM) architecture in forecasting time series of stock prices in the Brazilian capital market. For this purpose, historical daily data were collected for the assets Petrobras (PETR4.SA), Vale (VALE3.SA), and Itaú (ITUB4.SA) over the period from 2005 to 2025. The methodology involved the use of advanced computational models to analyze complex financial systems and included, as part of the data preprocessing, normalization using the MinMaxScaler function and structuring the data into 60-day time windows to feed a deep LSTM model. This model consists of four layers and was regularized using the Dropout technique (with a rate of 0.2) to prevent overfitting. The results on the test set showed varied performance: high accuracy was achieved for the banking sector asset (Itaú), good trend forecasting ability for the oil sector asset (Petrobras), and a quantitatively significant failure for the mining sector asset (Vale), evidenced by a notable error and a "cluster" at the end of the training epoch, occurring only for the Vale asset. It is concluded that, although LSTM networks show great potential for modeling financial time series, their effectiveness strongly depends on the particular characteristics and stability of each asset. This reinforces the idea that the financial market is a complex and non-universal system.

Keywords: Time series, Econophysics, Artificial Neural Networks.

LISTA DE FIGURAS

Figura 2.1 – Neurônio artificial.	16
Figura 3.1 – Rede perceptron multicamadas.	18
Figura 3.2 – Configuração de neurônio utilizado na derivação do algoritmo backpropagation	19
Figura 4.1 – Rede de Hopfield convencional.	22
Figura 4.2 – Ilustração de pontos de equilíbrio.	23
Figura 5.1 – Ilustração da rede recorrente de Jordan.	27
Figura 5.2 – Ilustração dos primeiros passos da rede LSTM.	28
Figura 5.3 – Ilustração da rede LSTM finalizada.	29
Figura 6.1 – Previsão do modelo (azul) vs. Preço real (vermelho) para ITUB4. Note a alta aderência da previsão ao seguir as flutuações do ativo.	33
Figura 6.2 – Comparativo entre o preço real da PETR4 e as previsões do modelo. A previsão acompanha a tendência, mas suaviza os picos e vales.	34
Figura 6.3 – Comparativo entre o preço real e a previsão para VALE3 em dias. O deslocamento vertical evidencia a falha do modelo em prever os preços.	35
Figura 6.4 – Série Temporal dos Retornos Logarítmicos Diários Sobrepostos (2005-2025)	35

SUMÁRIO

INTRODUÇÃO	12
1 INTRODUÇÃO ÀS SÉRIES TEMPORAIS	13
1.1 Definição e Conceitos Básicos	13
2 REDES NEURAIS ARTIFICIAIS: FUNDAMENTOS E ARQUITETURAS	15
2.1 Fundamentos de Redes Neurais Artificiais	15
2.2 Arquiteturas de Redes Neurais Artificiais	17
3 REDES PERCEPTRON MULTICAMADAS(PMC) E PROCESSO DE TREINAMENTO	18
3.1 Rede perceptron multicamadas	18
3.2 Processo de treinamento	19
4 REDES RECORRENTES E DE HOPFIELD	21
4.1 Introdução a Redes Recorrentes e de Hopfield	21
4.2 Funcionamento da rede de Hopfield	24
5 REDES RECORRENTES: REDES DE JORDAN E REDES LSTM	26
5.1 Redes de Jordan	26
5.2 Rede Long Short-Term Memory (LSTM)	27
6 METODOLOGIA, RESULTADOS E DISCUSSÕES	30
6.1 Metodologia	30
6.2 Resultados e discussões	32
6.2.1 Ativo ITUB4.SA (Itaú Unibanco)	32
6.2.2 Ativo PETR4.SA (Petrobras)	33
6.2.3 Ativo VALE3.SA(VALE): Análise de uma Falha Preditiva	34
CONCLUSÃO	37
REFERÊNCIAS	38

INTRODUÇÃO

O mercado financeiro, com suas flutuações e comportamentos imprevisíveis, pode ser visto como um sistema complexo, muito parecido com sistemas físicos. A área da Econofísica busca aplicar ferramentas teóricas da física para entender fenômenos econômicos. Um dos pontos que trouxe motivação a esse estudo foi a seguinte pergunta: Seria possível criar um modelo computacional capaz de aprender os padrões históricos de uma série temporal financeira para prever seu comportamento futuro? Com o objetivo de tentar encontrar respostas a essa pergunta, começamos a desenvolver e avaliar um modelo baseado em Redes Neurais Artificiais para prever o preço de fechamento de ações da bolsa brasileira. Isso nos levou a adentrar no campo de Séries Temporais e Redes Neurais Recorrentes, com o objetivo de verificar se de fato conseguiríamos prever o preço de três ações do mercado financeiro.

1 INTRODUÇÃO ÀS SÉRIES TEMPORAIS

Série temporal é uma sequência de observações de uma variável ao longo do tempo. Normalmente, uma tal “ordenação cronológica” é feita a intervalos regulares. Sendo assim, por construção, torna-se possível analisar padrões, tendências e comportamentos da variável ao longo do tempo, tanto para se abstrair conhecimento a respeito do fenômeno, quanto prever comportamentos futuros.

1.1 DEFINIÇÃO E CONCEITOS BÁSICOS

Uma série temporal é caracterizada por "uma sequência de dados registrados em intervalos de tempo sucessivos", onde cada valor Z_t de uma determinada grandeza está associado a um instante t . (BOX; JENKINS; REINSEL; LJUNG, 2015, p. 13)

O principal objetivo do estudo de séries temporais é "compreender a estrutura dos dados e prever valores futuros com base em informações históricas". (HYNDMAN; ATHANASOPOULOS, 2018, p. 5)

Os principais componentes de uma série temporal são:

- Tendência: "Uma direção geral de crescimento ou declínio que se manifesta ao longo do tempo". (CHATFIELD, 2003, p. 23)
- Sazonalidade: "Oscilações regulares e previsíveis que se repetem em intervalos de tempo fixos, como ciclos anuais". (HYNDMAN; ATHANASOPOULOS, 2018, p. 10)
- Ciclos: "Variações que ocorrem em períodos irregulares, geralmente influenciadas por fatores econômicos ou sociais". (BROCKWELL; DAVIS, 2016, p. 19)
- Aleatoriedade: "Flutuações imprevisíveis ou ruídos que não seguem um padrão identificável". (BOX; JENKINS; REINSEL; LJUNG, 2015, p. 45)

Considere uma série temporal $Z_t, t = 1, \dots, N$. Essa série temporal pode ser decomposta em três componentes não observáveis, conforme mostra equação 1:

$$Z_t = T_t + S_t + a_t \quad (1)$$

em que T_t e S_t representam a tendência e a sazonalidade, respectivamente e a_t é uma componente aleatória, de média zero e variância constante σ_a^2 . Quando se analisa uma série temporal da forma apresentada na equação 1, o interesse principal será obter cada componente partindo da hipótese de que as componentes sejam independentes uma da outra. (MORETTIN; TOLOI, 2006, p. 51)

Segundo Morettin e Toloi (2006) os três métodos mais utilizados para analisar a tendência são:

- Tendência polinomial: consiste em ajustar uma curva polinomial do tempo;
- Suavização ao redor de um ponto: consiste em suavizar os valores ao redor de um ponto, para estimar a tendência naquele ponto.
- Suavização através de ajustes: consiste em suavizar os valores através de sucessivos ajustes de retas de mínimos quadrados ponderados ("lowess").

Os métodos de análise de tendência mencionados, assim como outros modelos estatísticos clássicos, são ferramentas robustas para a decomposição e análise de séries temporais. Contudo, eles frequentemente partem de pressupostos de linearidade ou de distribuições estatísticas específicas, apresentando limitações ao lidar com as componentes cíclicas, não lineares e a aleatoriedade complexa (ruído) inerentes a sistemas dinâmicos como o mercado financeiro.

Diante desse desafio, as Redes Neurais Artificiais (RNAs) surgem como uma alternativa poderosa. Por sua capacidade de aprender padrões complexos e relações não lineares, sem a necessidade de um modelo matemático pré-definido, as RNAs são especialmente adequadas para a modelagem e previsão de séries temporais com as características do mercado de ações.

2 REDES NEURAIS ARTIFICIAIS: FUNDAMENTOS E ARQUITETURAS

2.1 FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS

Esta seção apresenta os principais conceitos de redes neurais artificiais, desde as principais arquiteturas feedforward até as recorrentes com ênfase na rede de Hopfield.

Introdução às Redes Neurais

O primeiro trabalho que envolvia modelos matemáticos inspirados em neurônios biológicos data de meados de 1940, por meio do artigo [McCulloch e Pitts \(1943\)](#).

A partir disso alguns trabalhos em Redes Neurais Artificiais (RNA) começaram a surgir, porém foi somente a partir de 1990 que houve um crescimento nessa área.

As aplicações que envolvem RNA são as mais variadas possíveis. Podemos citar:

- Classificação de padrões de escrita e de fala;
- Previsão de ações do mercado financeiro;
- Identificação de anomalias em imagens médicas;
- E muitas outras.

Estrutura e Funcionamento das Redes Neurais

Podemos definir as redes neurais da seguinte forma:

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso dos seres vivos. Possuem a capacidade de aquisição e manutenção do conhecimento (baseado em informações) e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões (sinapses artificiais), sendo representadas aqui por vetores/matrizes de pesos sinápticos. (SILVA; SPATTI; FLAUZINO, 2016, p. 24).

Em uma representação visual, como na figura (2.1), os sinais de entrada ($x_1, x_2, x_3, \dots, x_n$) são vistos analogamente como os impulsos elétricos sendo recebidos por um neurônio real através de seus dendritos. As ponderações que ocorrem pelas junções sinápticas são representadas pelo conjunto de pesos sinápticos ($w_1, w_2, w_3, \dots, w_n$). Dessa forma, a cada sinal x_n tem-se um peso sináptico w_n associado. Os sinais ponderados pelos pesos são recebidos pelo neurônio e agregados pelo combinador linear (\sum). O limiar de ativação (θ) estabelece um limite inferior para o potencial de ativação, acima do qual ocorre um disparo em direção à saída do neurônio. A função de ativação ou potencial de ativação

(g) é obtido através da diferença do valor produzido pelo combinador linear e o limiar de ativação. A função de ativação (g) limita a saída do neurônio dentro de um intervalo de valores assumidos pela própria imagem da função. O sinal de saída(y) é o valor final ou a resposta final produzida pelo neurônio em relação ao conjunto de sinais de entrada.

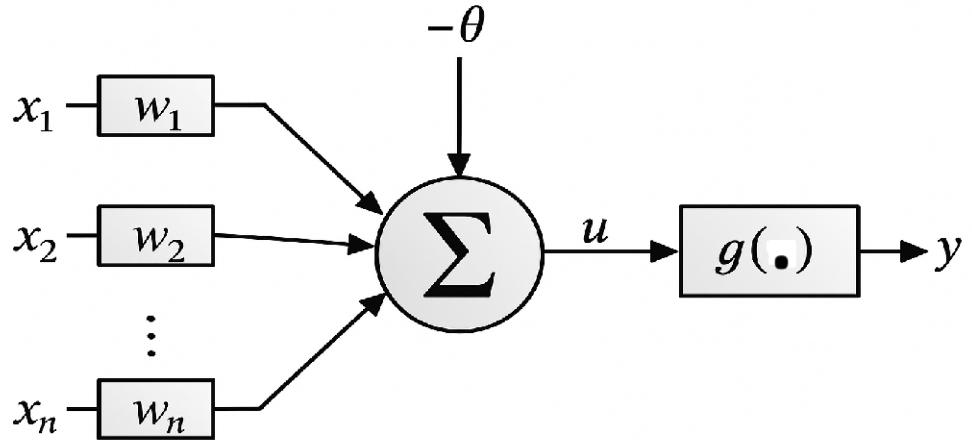


Figura 2.1 – Neurônio artificial.

Fonte: ([SILVA; SPATTI; FLAUZINO, 2016, p. 34](#))

A saída do neurônio pode ser escrita como:

$$y = g(u) \quad (2)$$

onde:

$$u = \sum_{i=1}^n w_i x_i - \theta \quad (3)$$

Os neurônios artificiais utilizados nos modelos de redes neurais artificiais são não-lineares, fornecem saídas tipicamente contínuas, e realizam funções simples, como coletar os sinais existentes em suas entradas, agregá-los de acordo com a sua função operacional e produzir uma resposta levando em consideração sua função de ativação inerente ([SILVA; SPATTI; FLAUZINO, 2016, p. 33](#)).

2.2 ARQUITETURAS DE REDES NEURAIS ARTIFICIAIS

Arquitetura versus topologia de Redes Neurais

Redes neurais são usadas para se obter “conhecimento” e solucionar uma vasta gama de problemas. Dois importantes atributos das redes, a saber, arquitetura e topologia, são de fundamental importância, visto que, a escolha adequada de cada um deles pode determinar o sucesso ou não de sua aplicação. Definamos, portanto, cada um deles.

Na arquitetura, nos preocupamos muito em como os neurônios estão arranjados entre si conforme pode ser visto no trecho:

A arquitetura de uma rede neural artificial define a forma como os neurônios estão arranjados, ou dispostos, uns em relação aos outros. Esses arranjos são essencialmente estruturados através do direcionamento das conexões sinápticas dos neurônios.([SILVA; SPATTI; FLAUZINO, 2016](#), p.45)

A topologia, por outro lado, se preocupa mais em mostrar as diferentes formas de composições estruturais de determinada arquitetura como mostra no trecho:

Já a topologia de uma rede neural, considerando determinada arquitetura, pode ser definida como sendo as diferentes formas de composições estruturais que esta poderá assumir. Por exemplo, podem-se ter duas topologias pertencentes a uma mesma arquitetura , sendo uma composta de 10 neurônios e a outra 20 neurônios; ou ainda uma constituída de neurônios com função de ativação logística , ao passo que os neurônios da outra assumem a tangente hiperbólica como função de ativação.([SILVA; SPATTI; FLAUZINO, 2016](#), p.45)

Principais arquiteturas de uma rede neural

Podemos dividir uma rede neural em 3 partes principais:

- camada de entrada;
- uma ou mais camadas escondidas ou intermediarias;
- camada de saída.

Quanto à arquitetura de uma rede, as principais podem ser divididas em: redes feedforward (alimentação para a frente) de camada simples, redes feedforward de camadas múltiplas, redes recorrentes e redes reticuladas.

3 REDES PERCEPTRON MULTICAMADAS(PMC) E PROCESSO DE TREINAMENTO

3.1 REDE PERCEPTRON MULTICAMADAS

Conhecidas como (PMC) e por sua grande versatilidade em aplicações, as redes perceptron multicamadas além da camada de entrada e a de saída, possuem pelo menos uma camada intermediária. Os estímulos são apresentados a rede em sua camada de entrada. Em seguida, as camadas intermediárias processam e extraem as informações modificando os pesos sinápticos e, por fim, os neurônios da camada de saída recebem esses estímulos advindos das camadas anteriores e produz uma resposta que será a saída da rede. Aqui temos uma arquitetura “feedforward”, ou seja, o processamento dos sinais apresentados à rede ocorre ao longo da direção da camada de entrada para a camada de saída, a princípio sem haver qualquer tipo de realimentação de valores como apresentado na figura(3.1).

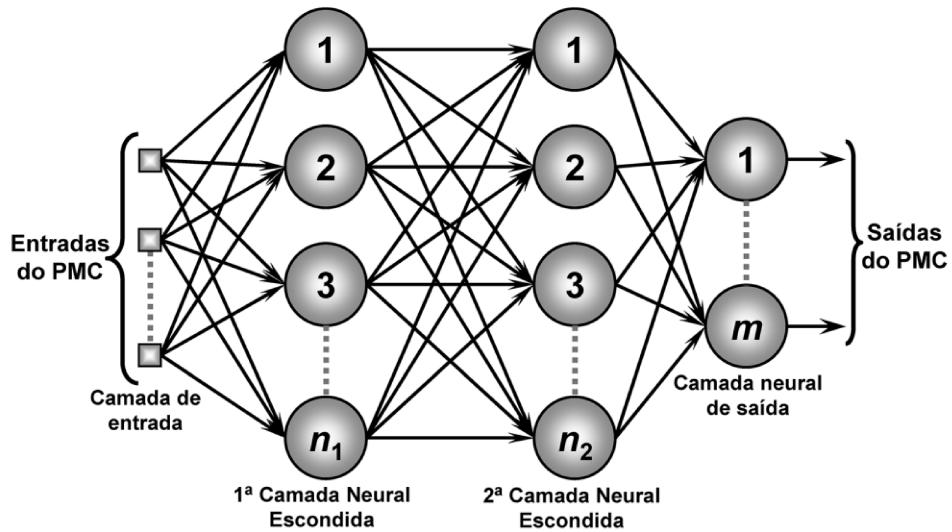


Figura 3.1 – Rede perceptron multicamadas.

Fonte: ([SILVA; SPATTI; FLAUZINO, 2016, p.92](#))

Dois pontos importantes precisam ser destacados. Em primeiro lugar, a rede é “fully connected”, isto é, cada neurônio de cada camada oculta está conectado com todos os outros neurônios das camadas adjacentes. Além disso, pode haver mais que um neurônio na camada de saída. Os pesos sinápticos mediando as conexões entre os neurônios são ajustados utilizando-se um processo de treinamento supervisionado. Neste processo, é apresentado à rede um conjunto de registros tão grande quanto possível. Pretende-se que a rede aprenda a relação entre uma quantidade previamente definida de atributos ou características destes registros (os quais serão os sinais de entrada) e um atributo meta (que pode ser uma classe nas tarefas de classificação e que será a saída ou resposta produzida

pela rede). Desta forma, de uma maneira iterativa, busca-se minimizar o erro entre o valor apresentado pela rede e o valor esperado que já é conhecido. Durante o treinamento, os pesos sinápticos são ajustados iterativamente e o treinamento termina quando o erro atinge um valor ótimo (mínimo). Neste ponto, dizemos que a rede “aprendeu” o relacionamento entre os atributos previsores e o atributo meta.

3.2 PROCESSO DE TREINAMENTO

O treinamento de uma Rede Perceptron Multicamadas (PMC) é um processo iterativo e faz uso do algoritmo Backpropagation, visando ajustar os pesos sinápticos e os bias da rede para minimizar a diferença entre a saída produzida pela rede e a saída desejada. Com esse intuito utiliza-se um conjunto de dados tão grande quanto possível. Este processo geralmente envolve duas etapas:

- Fase forward: As entradas são apresentadas à rede e propagadas através das camadas. Cada neurônio calcula sua saída com base nas entradas ponderadas da camada anterior e na função de ativação. A saída apresentada é utilizada como entrada para os neurônios da camada subsequente, até que a saída final da rede seja gerada.

Após a propagação direta, a saída gerada pela rede é comparada com a saída esperada. A diferença entre essas duas saídas é quantificada por uma função de erro (ou função de custo), como o erro quadrático médio (MSE - Mean Squared Error). O objetivo é reduzir este erro ao mínimo possível.

A figura 3.2 exemplifica um neurônio utilizado na derivação do algoritmo backpropagation com alguns parâmetros importantes.

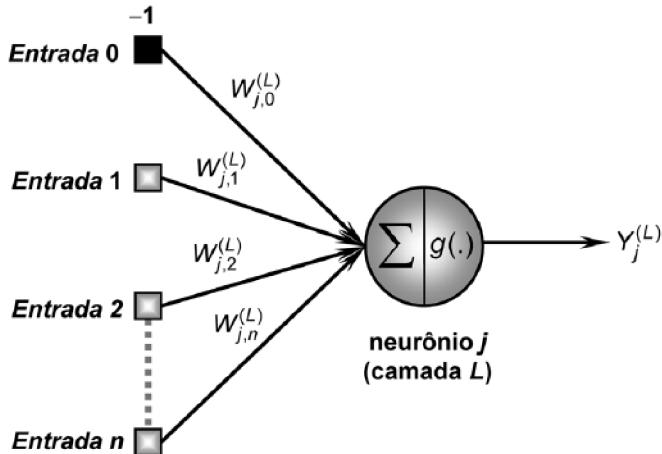


Figura 3.2 – Configuração de neurônio utilizado na derivação do algoritmo backpropagation

Fonte: (SILVA; SPATTI; FLAUZINO, 2016, p.96)

Os erros quadrático $E(k)$ e quadrático médio E_M são definidos da seguinte forma:

$$E(k) = \frac{1}{2} \sum_{j=1}^{n_L} (d_j(k) - y_j^{(L)}(k))^2 \quad (4)$$

$$E_M = \frac{1}{p} \sum_{k=1}^p E(k) \quad (5)$$

O objetivo do processo de treinamento consiste em ajustar as matrizes de pesos $W_{ji}^{(L)}$ a fim de minimizar o erro entre a saída produzida e a desejada. ([SILVA; SPATTI; FLAUZINO, 2016](#))

- Fase backward: O erro calculado na camada de saída é propagado para trás através da rede, da camada de saída em direção à camada de entrada. Durante a retropropagação, o algoritmo calcula o gradiente da função de erro ou função de custo em relação a cada peso sináptico (conectando cada par de neurônios em camadas adjacentes) e bias (de cada neurônio na rede). Esse gradiente, de certa maneira, indica a direção e a magnitude do ajuste necessário para diminuir o erro.

As etapas de propagação direta, cálculo do erro, retropropagação e ajuste de pesos são repetidas para todo o conjunto de dados de treinamento por um determinado número de épocas (uma época nada mais é do que apresentação de todo o conjunto de dados à rede seguida da propagação direta e subsequente retropropagação) ou até que o erro atinja um limite aceitável ou pare de diminuir significativamente.

Segundo [Silva, Spatti e Flauzino \(2016\)](#) a esse limite aceitável damos o nome de critério de parada que leva em consideração todas as amostras de treinamento disponíveis, podendo ser definido como:

$$|E_M^{\text{atual}} - E_M^{\text{anterior}}| \leq \varepsilon \quad (6)$$

Para garantir que a rede não está apenas memorizando os dados de treinamento (overfitting), é comum dividir os dados disponíveis em conjuntos de treinamento, validação e teste. O conjunto de validação é usado para monitorar o desempenho da rede durante o treinamento e ajustar hiperparâmetros, enquanto o conjunto de teste é usado para avaliar o desempenho final da rede em dados nunca antes vistos.

Este processo iterativo permite que a Rede Perceptron Multicamadas "aprenda" os padrões e as relações complexas presentes nos dados, tornando-a capaz de fazer previsões ou classificações precisas em novos dados.

4 REDES RECORRENTES E DE HOPFIELD

4.1 INTRODUÇÃO A REDES RECORRENTES E DE HOPFIELD

O campo das redes neurais artificiais experimentou um período de grande entusiasmo até meados de 1969, quando o interesse diminuiu drasticamente.

Foi somente em 1982 com os trabalhos do físico John Hopfield ([HOPFIELD, 1982](#)) que uma nova perspectiva começou a ressurgir para o campo. O trabalho sobre Redes de Hopfield foi um divisor de águas e contribuiu significativamente para o renascimento de importantes pesquisas que estavam estagnadas.

O que são redes recorrentes?

Para entendermos como funciona uma rede recorrente façamos uso de um exemplo. Imagine que você se propõe a assistir um filme. Para entender a cena atual, você não precisa apenas ver o que está acontecendo agora, mas também lembrar o que aconteceu nas cenas anteriores. Se você esquecesse tudo a cada novo momento, a história não faria sentido. Em redes recorrentes a ideia é análoga: é preciso lembrar do passado para compreender o presente.

Vimos que nas redes neurais tradicionais, como as “feedforward” a informação flui da camada de entrada para a de saída em uma única direção e o processamento de cada entrada ocorre de forma independente. Por outro lado uma Rede Neural Artificial Recorrente(RNA Recorrente), é um tipo de arquitetura em que as saídas podem ser realimentadas às suas próprias entradas ou para camadas anteriores criando um ciclo de “feedback”. Dessa forma ela se destaca por:

- Lidar com informações que dependem de um histórico, como texto, fala ou séries temporais.
- Os estados internos de uma rede e suas saídas passadas influenciam seu estado e saída atuais.

O que são redes recorrentes de Hopfield?

As Redes Recorrentes de Hopfield, como esquematizadas na figura [4.1](#), são um tipo de rede neural artificial que se destaca por sua capacidade de armazenar e recuperar padrões, funcionando como uma memória associativa. O grande marco de Hopfield foi a capacidade de mostrar que redes recorrentes de uma única camada podem ser caracterizadas por uma função de energia relacionada aos estados de seu comportamento dinâmico.

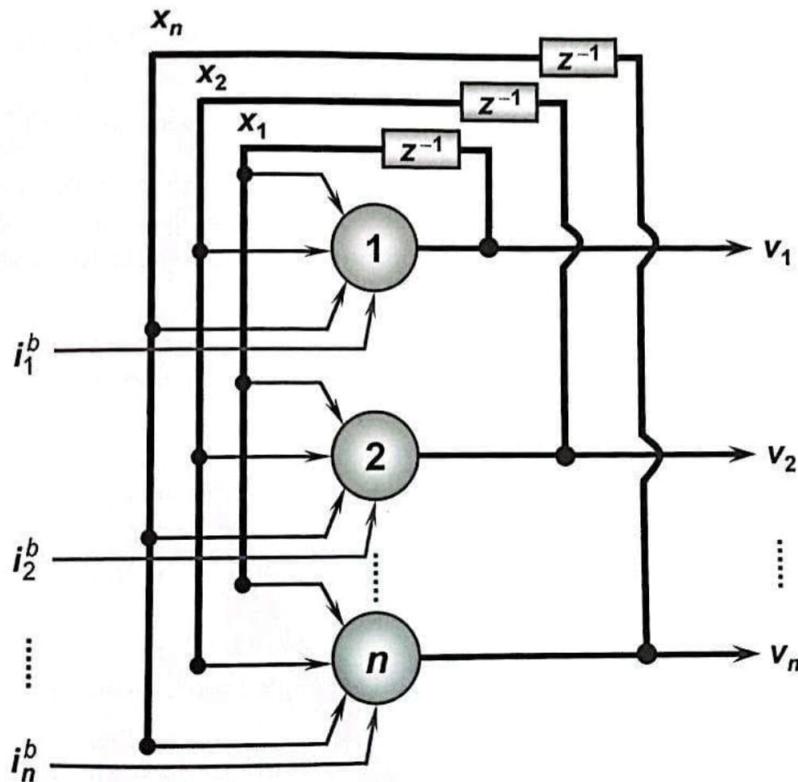


Figura 4.1 – Rede de Hopfield convencional.

Fonte: (SILVA; SPATTI; FLAUZINO, 2016, p.201)

Essas redes possuem uma arquitetura com realimentação global, ou seja, as saídas de todos os neurônios podem influenciar as entradas de todos os outros neurônios da rede. A rede em si não possui uma saída instantânea e estática, seu estado interno evolui ao longo do tempo, buscando um estado de estabilidade. Além disso, ela é capaz de “aprender” e “armazenar” padrões, de tal forma que se futuramente for apresentada uma versão parcial ou ruidosa de algum desses padrões, ela pode tentar recuperar o padrão original, esse funcionamento é conhecido como princípio das memórias associativas.

John Hopfield mostrou que o comportamento dinâmico de uma rede recorrente pode ser mapeado utilizando o conceito de minimização de uma função de energia, frequentemente denotada por $E(x)$.

Imagine um terreno montanhoso como ilustrado na figura(4.2), em que o eixo vertical $E(x)$ representa o valor da energia e o eixo x horizontal representa o “estado” da rede(ou seja, as configurações das saídas dos neurônios).

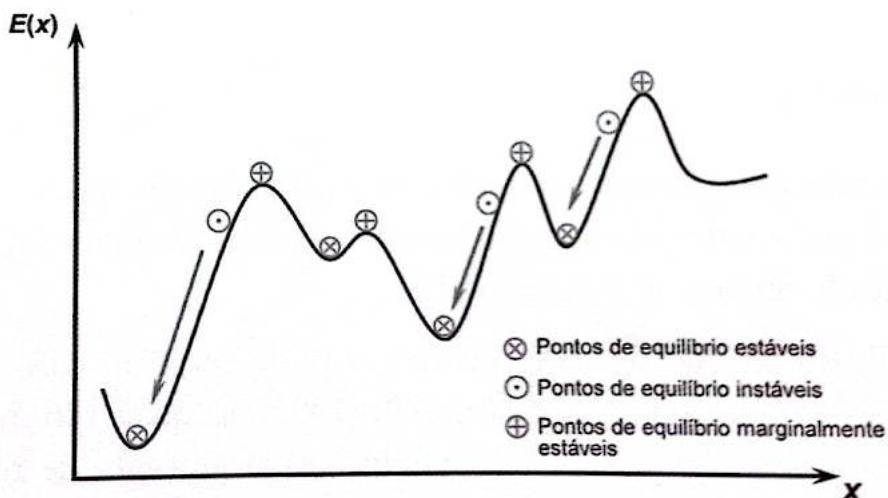


Figura 4.2 – Ilustração de pontos de equilíbrio.

Fonte: ([SILVA; SPATTI; FLAUZINO, 2016](#), p.200)

O objetivo da rede é reduzir a sua energia, como uma bola que rola ladeira abaixo no terreno. Essa “descida” continua até que a bola atinja um ponto onde não consegue mais descer, esses pontos são chamados de pontos de equilíbrio da rede.

Existem diferentes tipos de pontos de equilíbrio:

- **Pontos de Equilíbrio Estáveis:** São os "vales" ou "poços" no terreno de energia. Quando a rede atinge um desses pontos, ela se estabiliza. Estes são os estados desejados, pois representam as soluções esperadas para um problema específico ou os padrões que foram memorizados. A tendência é que a rede sempre se mova em direção a um desses pontos de equilíbrio estáveis.
- **Pontos de Equilíbrio Instáveis:** São os "picos" das montanhas. Se a rede parar em um pico, qualquer pequena perturbação a fará "rolar" para longe.
- **Pontos de Equilíbrio Marginalmente Estáveis:** São como "platôs" ou "encostas planas". A rede pode permanecer ali por um tempo, mas não é um ponto de parada definitivo.

Dessa forma, a rede de Hopfield busca minimizar a função de energia, e essa minimização leva aos pontos de equilíbrio estáveis, que são os padrões aprendidos pela rede ou soluções esperadas. ([SILVA; SPATTI; FLAUZINO, 2016](#))

Aplicações gerais da rede de Hopfield

As principais aplicações das Redes de Hopfield envolvem a área de otimização de sistemas. Isso se deve à sua natureza de minimização de energia, que pode ser mapeada para problemas onde se busca a melhor solução possível. Podemos citar como exemplos:

- Programação Dinâmica: Problemas onde as decisões futuras dependem das decisões tomadas no passado.
- Programação Linear: Otimização de uma função objetivo linear sujeita a restrições lineares.
- Otimização Não-Linear Restrita: Problemas mais complexos com relações não-lineares e condições de contorno (restrições).
- Otimização Combinatória: Encontrar a melhor combinação de itens ou ações entre um grande número de possibilidades. O exemplo clássico desta categoria é o "Problema do Caixeiro-Viajante"(PCV), que, de forma análoga ao "problema da melhor rota de entrega", busca o caminho mais curto para visitar diversos pontos e retornar à origem. A aplicação bem-sucedida das redes de Hopfield para solucionar o PCV, demonstrada por [Hopfield e Tank \(1985\)](#), foi um marco que solidificou o potencial da arquitetura para problemas de otimização.

4.2 FUNCIONAMENTO DA REDE DE HOPFIELD

Arquitetura: Simplicidade e Conectividade Total

A rede de Hopfield originalmente, apresenta uma arquitetura simples, sendo constituída de uma única camada, em cada neurônio está conectado a todos os outros e a si mesmo(para certas aplicações, como memórias associativas, a auto-realimentação é tipicamente desativada). Essa interconexão global significa que as saídas de todos os neurônios realimentam as entradas de todos os outros neurônios da rede. Exemplificando, pensemos em uma roda de conversa, em que várias pessoas estão discorrendo sobre um determinado assunto. A conversa coletiva evolui até que todos cheguem a um “consenso”(o ponto de equilíbrio).

Breve descrição da dinâmica

Segundo [Silva, Spatti e Flauzino \(2016\)](#) o comportamento dinâmico de uma Rede de Hopfield pode ser elucidado como uma trajetória em seu espaço de estados, discretizada em instantes de tempo. A evolução do estado da rede obedece à seguinte sequência:

- **No instante inicial (t_0):** Um vetor de entrada externo $x(t_0)$ é aplicado à rede, gerando como resposta um vetor de saída inicial $v(t_0)$.

- **Nos instantes subsequentes** (t_1, t_2, \dots): Para cada passo de tempo t_k (onde $k \geq 1$), o estado da rede é atualizado de forma recorrente. A entrada para o cálculo corrente, $x(t_k)$, é definida como a saída da iteração imediatamente anterior, ou seja, $x(t_k) = v(t_{k-1})$. Este processo gera uma nova saída, $v(t_k)$.
- **Condição de Estabilização para** (t_k): O processo iterativo de atualização de estados continua até que a rede converja para um estado estável. A condição de estabilidade, atingida no instante t_k , é satisfeita quando uma saída se torna idêntica à anterior, isto é, $v(t_k) = v(t_{k-1})$. Este estado final representa um mínimo local na função de energia da rede e corresponde ao padrão memorizado associado ao vetor de entrada inicial.

Para que um sistema dinâmico, como a Rede de Hopfield seja útil em aplicações como memórias associativas ou problemas de otimização, é fundamental que seu comportamento seja estável, ou seja, que ela sempre converja para uma solução e não oscile indefinidamente ou entre em caos.

Embora a aplicação prática deste trabalho se concentre na arquitetura LSTM(Long Short-Term Memory) que veremos mais a frente, um estudo aprofundado da Rede de Hopfield é apresentado por seu valor seminal no campo das redes recorrentes. Os conceitos de dinâmica de estados, minimização de uma função de energia e estabilidade, formalizados por Hopfield, são pilares que fundamentam o comportamento de arquiteturas recorrentes mais modernas e, dessa forma, sua compreensão é essencial.

5 REDES RECORRENTES: REDES DE JORDAN E REDES LSTM

O progresso tecnológico impulsionou o monitoramento e o interesse no mercado de ações, levando especialistas a buscarem métodos automáticos para analisar as flutuações do mercado. Contudo, prever o comportamento dos títulos financeiros continua sendo um desafio que intriga tanto especialistas quanto investidores. Nesse contexto, as redes recorrentes surgem como uma alternativa promissora para auxiliar na tomada de decisões nesse mercado.

Segundo [Silva, Spatti e Flauzino \(2016\)](#) os dois principais modelos que guiam as estratégias de compra e venda de ações são:

- **Modelo fundamentalista:** analisa os aspectos econômicos e contábeis das empresas. Considera também as influências de variáveis macroeconômicas, como taxas de juros, inflação, crescimento do PIB, entre outros.
- **Modelo Técnico:** baseia-se em dados históricos a respeito da empresa.

Vale ressaltar que esses dois modelos são complementares e nos ajuda a entender como podemos utilizar as redes recorrentes nesse contexto.

Ao analisar os gráficos de evolução de preços no mercado financeiro, percebemos que seu comportamento é altamente não linear, tendendo a um estado caótico, o que dificulta a aplicação de métodos estatísticos e econômicos tradicionais para capturar as relações complexas. As redes recorrentes, com sua capacidade de aprender e modelar relações complexas e não lineares, tornam-se ideais para essa tarefa.

As redes neurais recorrentes também são utilizadas para capturar dependências temporais. Isso ocorre porque suas saídas anteriores são reintroduzidas como entradas, influenciando o estado atual da rede. Essa particularidade ressalta a relevância de seu uso, visto que o valor de uma ação na atualidade é condicionado não só pelo preço do dia anterior, mas por uma sucessão de eventos passados. Além disso elas são adequadas para prever múltiplos passos à frente em uma série temporal.

Diversos estudos demonstraram a eficácia das redes recorrentes na previsão de séries temporais com comportamentos não lineares como o mercado de ações. As redes mais utilizadas são as de Jordan e a LSTM, das quais falaremos um pouco nas próximas seções. Contudo, dada a simplicidade e limitação das redes de Jordan, preferimos usar, neste trabalho, a rede LSTM.

5.1 REDES DE JORDAN

As redes de Jordan são caracterizadas por ter **uma conexão de realimentação** da camada de saída para a camada de entrada. A figura [5.1](#) ilustra sua estrutura.

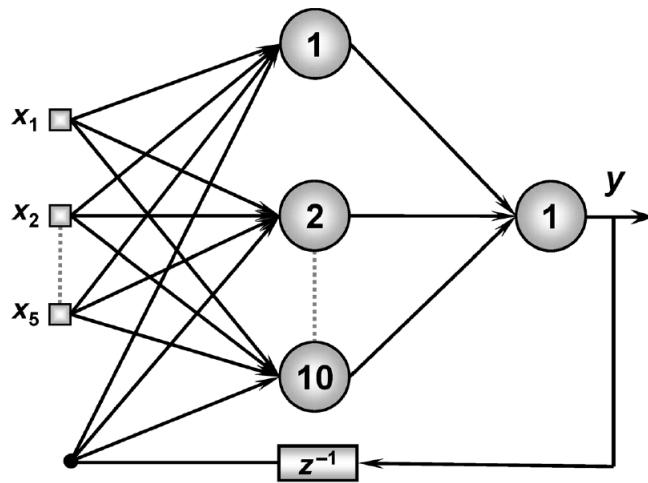


Figura 5.1 – Ilustração da rede recorrente de Jordan.

Fonte: ([SILVA; SPATTI; FLAUZINO, 2016, p.303](#))

Como ilustrado na figura 5.1 o processamento da rede de Jordan é como segue: a camada de entrada recebe os dados iniciais, a camada escondida processa a informação e a camada de saída produz o resultado da previsão da rede, cujo valor é retroalimentado para a camada de entrada. Pode-se aprimorar a previsão empregando-se múltiplas redes de Jordan, incorporando variáveis adicionais relevantes. Por exemplo, uma rede poderia processar dados históricos de preço e volume, enquanto outra se dedicaria a variáveis macroeconômicas.

Estudos recentes demonstram a eficácia dessa arquitetura na modelagem de fenômenos econômicos. Por exemplo, [HOQUE e ALJAMAAN \(2021\)](#) aplicaram uma Rede de Jordan à previsão da inflação na Zona do Euro e observaram desempenho comparável e em alguns casos superior ao de previsores humanos e a modelos econôméticos tradicionais. Esse resultado reforça a relevância histórica e prática das redes de Jordan como precursoras de arquiteturas mais sofisticadas, como as LSTM, destacando sua contribuição para o avanço das técnicas de previsão em sistemas econômicos complexos.

5.2 REDE LONG SHORT-TERM MEMORY (LSTM)

As redes recorrentes tradicionais, como a de Jordan, são eficazes para analisar informações em um contexto temporal curto. No entanto, o problema surge quando é necessário um contexto maior ou informações por longos períodos. Imagine que você está escrevendo um texto de dez páginas e no primeiro parágrafo, informa que tem um irmão gêmeo. A partir daí, o texto segue normalmente, sem fazer mais nenhuma referência a isso. No final do texto, você pergunta ao leitor se possui uma pessoa com fisionomia parecida com a sua e que é seu parente. Uma rede neural tradicional teria dificuldade em lembrar de tantas palavras, principalmente porque o contexto-chave está no início do texto. Dessa forma, a lacuna entre a informação relevante e o ponto em que ela é necessária é muito

grande. À medida que essa lacuna aumenta, as redes recorrentes tradicionais se tornam ineficazes.

Tecnicamente, essa dificuldade é conhecida como o problema do gradiente evanescente (vanishing gradient problem). Em sequências longas, o gradiente do erro, que é propagado para trás no tempo durante o treinamento, pode diminuir exponencialmente até se tornar tão pequeno que os pesos das camadas iniciais da rede não são mais atualizados de forma significativa, impedindo o aprendizado de dependências de longo prazo (BENGIO; SIMARD; FRASCONI, 1994).

Com o intuito de resolver esse problema, Hochreiter e Schmidhuber(1997) elaboraram uma arquitetura conhecida como Long Short-Term Memory (LSTM).

Uma rede LSTM é dividida em células(A) e possui primeiramente um conjunto de dados chamados de estado anterior da célula(C_{t-1}) que flui através da rede com poucas transformações lineares. A capacidade de adicionar informações ou novos contextos são controlados por estruturas chamadas de portas(chaves). As portas são formadas por uma camada de rede neural com função de ativação sigmoide e uma operação de multiplicação ponto a ponto. A função sigmoide comprime os valores de saída entre 0 e 1, definindo a quantidade de informação que deve passar: um valor próximo de 0 significa "não deixar passar nada", enquanto um valor próximo de 1 significa "deixar passar tudo", a figura 5.2 ilustra esse mecanismo (OLAH, 2015).

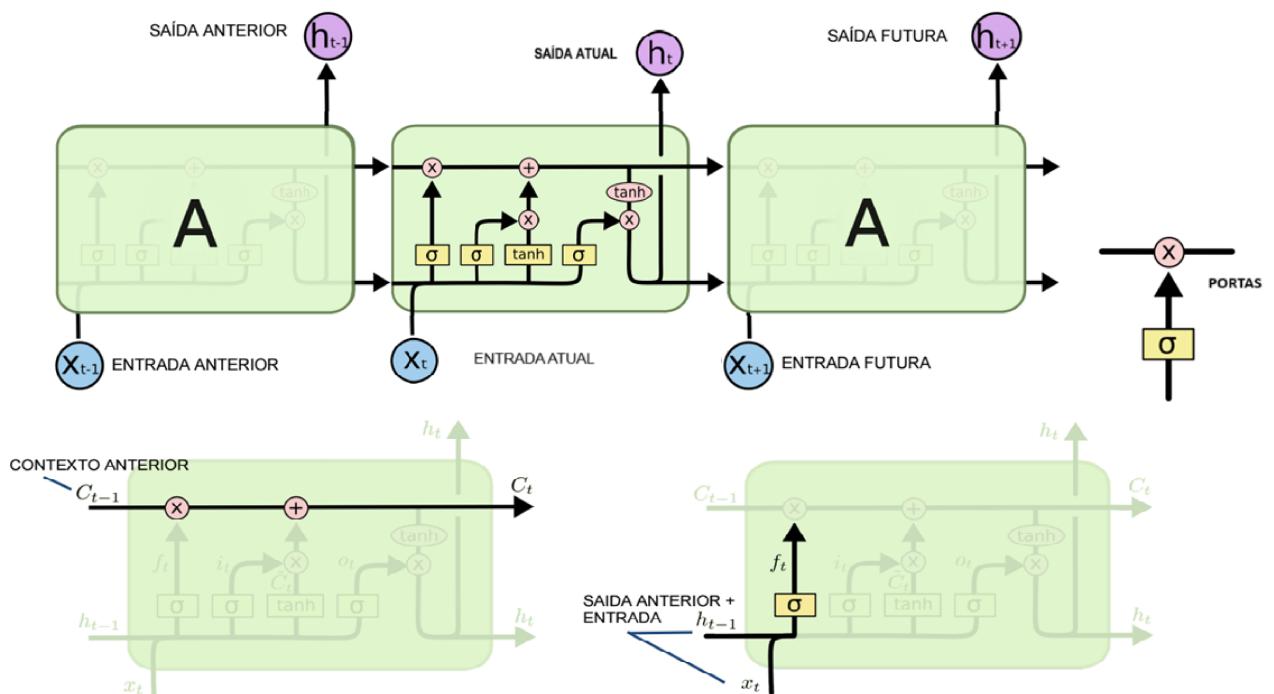


Figura 5.2 – Ilustração dos primeiros passos da rede LSTM.

Fonte: (OLAH, 2015)

A atualização da memória da célula, de C_{t-1} ocorre em duas etapas principais:

- Porta de Esquecimento(Forget Gate): Primeiramente, o portão de esquecimento f_t analisa a memória anterior C_{t-1} e decide quais informações devem ser descartadas.
- Porta de Entrada/Atualização(Input Gate): Em paralelo, uma camada de rede neural gera um vetor com valores candidatos \tilde{C}_t para serem adicionados à memória. O portão de entrada i_t atua como um filtro, decidindo a importância de cada um desses valores. Apenas as informações que passam por esse filtro são combinadas com a memória restante para formar o novo estado da célula, C_t .
- Porta de saída(Output Gate): Esta porta filtra o estado da célula C_t para determinar qual informação será enviada como o estado oculto h_t , que também serve como a saída da rede para o passo de tempo atual.

Vale ressaltar que o estado da célula C_t , que contém a memória de longo prazo atualizada, é passado por uma função tanh para escalar os valores entre -1 e 1 evitando uma oscilação caótica dos dados. A figura 5.3 ilustra a finalização do processo.

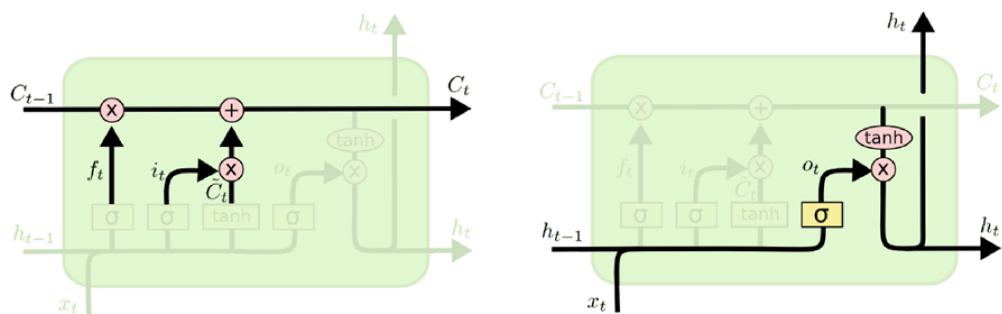


Figura 5.3 – Ilustração da rede LSTM finalizada.

Fonte: ([OLAH, 2015](#))

Graças a esse mecanismo de portas, a LSTM consegue reter informações por longos períodos e descartá-las quando não são mais relevantes, superando a principal limitação das RNNs tradicionais([HOCHREITER; SCHMIDHUBER, 1997](#)).

A capacidade de escolher como uma chave o que importa, torna as LSTM uma ferramenta extremamente poderosa para tarefas de previsão de séries temporais, como a modelagem do mercado de ações, onde o comportamento atual dos preços pode ser influenciado por uma complexa sequência de eventos passados.

6 METODOLOGIA, RESULTADOS E DISCUSSÕES

6.1 METODOLOGIA

Este capítulo descreve os procedimentos utilizados para o desenvolvimento do estudo da análise de séries temporais aplicadas ao mercado financeiro.

Foi utilizada a linguagem de programação python versão 3.12, com auxílio das bibliotecas mais utilizadas para análise de dados e aprendizado de máquina como: Pandas(para tabulação de dados), Scikit-learn(para padronização/normalização dos registros e cálculo de algumas estatísticas) e a API de alto nível Keras com o TensorFlow como Backend, para a construção do modelo de deep learning.

Extração dos dados e pré-processamento

O estudo focou na análise e previsão do preço de fechamento ajustado de três ativos mais representativos da bolsa de valores brasileira:

- Itaú Unibanco(ITUB4.SA);
- Petrobras(PETR4.SA);
- Vale(VALE3.SA).

Com o intuito de garantir que o modelo seja exposto a diferentes condições do mercado foi escolhido um horizonte temporal extenso. Foi definido um período histórico de 01/01/2005 a 31/07/2025.

Esses dados foram extraídos utilizando a biblioteca `yfinance`, ferramenta que permite uma interface direta com o portal Yahoo Finance. Para cada dia de previsão, foram coletadas as seguintes variáveis: Preço de abertura(Open), Preço Máximo(High), Preço mínimo(Low), Preço de fechamento(Close), volume de negociação(Volume) e Preço de fechamento ajustado(Adj Close).

Após a coleta, foi realizado um pré-processamento dos dados. Verificou-se que não havia dados ausentes(null), porém foram identificados registros em que o volume de ações era nulo. Registros assim aparecem devido a dias sem negociação ou erros na base de dados que se não fossem tratados poderia comprometer o processo de aprendizagem do modelo. Dados com volumes nulos, foram preenchidos com um valor correspondente ao volume médio. Verificou-se ainda que não houve registros com valores negativos.

Divisão da base de dados

Para cada ativo, o conjunto de dados foi dividido em duas partes: conjunto de treinamento e conjunto de teste. As primeiras 5000 amostras cronológicas foram reservadas para o treinamento, enquanto que 117 amostras subsequentes foram reservadas para o

teste. Essa abordagem garante que o modelo seja treinado com dados passados, e com isso, sua capacidade preditiva pode ser avaliada usando dados futuros, os quais nunca foram vistos antes.

Os preços das ações foram normalizados para o intervalo $[0,1]$ utilizando a função `MinMaxScaler` da biblioteca Scikit-learn. A normalização garante que todas as entradas tenham a mesma escala, melhorando a estabilidade do gradiente descendente e acelerando a convergência do modelo.

A arquitetura LSTM é ideal para aprender a partir de sequências. Os dados foram estruturados com uma abordagem conhecida como janela deslizante(sliding window). Uma janela de 60 passos de tempo(timesteps) foi utilizada para prever o preço de uma ação em um determinado dia t , o modelo utiliza como dados de entrada os preços normalizados dos 60 dias anteriores(de $t-60$ a $t-1$). O valor de saída que o modelo deve aprender a prever é o preço normalizado no dia t .

Arquitetura e Treinamento do Modelo LSTM

Foi utilizado um modelo Sequencial empilhado(Stacked LSTM), composto pelas seguintes camadas:

- **Quatro camadas LSTM:** Cada camada possui 50 neurônios sendo que as três primeiras camadas foram configuradas com `return_sequences=True`, permitindo que a saída de uma camada alimentasse a camada seguinte com a sequência completa viabilizando o empilhamento de camadas. A última camada LSTM foi configurada com `return_sequences=False`, produzindo apenas a saída correspondente ao último passo temporal.
- **Quatro camadas Dropout:** Após cada camada LSTM, foi inserida uma camada Dropout com uma taxa de 0.2(20%). Essa camada desativa aleatoriamente uma fração dos neurônios durante o treinamento, prevenindo o overfitting (superajustamento) e melhorando a capacidade de generalização do modelo.
- **Camada de saída:** Uma camada Dense final com uma única unidade de ativação linear foi utilizada para gerar a saída da previsão, resultando em um valor escalar contínuo.

O modelo foi otimizado utilizando o algoritmo ADAM (Adaptive Moment Estimation), muito utilizado em redes neurais. Ele é um método de gradiente estocástico que combina duas ideias principais: 1) mantém uma média móvel dos gradientes passados, ou seja, ele “lembra” a direção do ajuste anterior. Isso acelera a convergência e reduz oscilações; 2) Mantém uma média móvel do quadrado dos gradientes permitindo que o ADAM adapte a taxa de aprendizado para cada parâmetro, dando passos menores para parâmetros com gradientes grandes e passos maiores para gradientes pequenos. Ao combinar momento e

ajuste adaptativo, o ADAM consegue convergir rapidamente e ser mais estável do que outros otimizadores. O desempenho do modelo, foi avaliado por meio do erro quadrático médio (Mean Squared Error - MSE) como métrica, permitindo quantificar o desvio entre os valores previstos e observados.

O treinamento do modelo foi realizado utilizando a seguinte condição de parada: caso o treinamento não reduzisse o MSE por até 30 épocas de treinamento consecutivas, o treinamento seria interrompido; contudo, estabelecemos que o treinamento duraria até 500 épocas de treinamento, no máximo. Estabelecer uma condição de parada permite que o modelo não fique sobre ou superajustado, permitindo uma melhor generalização.

Avaliação de desempenho do modelo

A performance do modelo foi avaliada quantitativamente no conjunto de teste. Após a geração das previsões, os valores normalizados foram revertidos à escala original através da transformação inversa do **MinMaxScaler**. A métrica de avaliação utilizada foi a Raiz do Erro Quadrático Médio (Root Mean Squared Error - RMSE), utilizada para medir a magnitude média dos erros de previsão da mesma unidade da variável original. O resultado do RMSE e a análise visual dos gráficos de previsão, servirão de base para as discussões sobre a eficácia do modelo.

6.2 RESULTADOS E DISCUSSÕES

Serão apresentados e analisados os resultados obtidos a partir da aplicação do modelo de Rede Neural Recorrente com arquitetura LSTM. A performance do modelo foi avaliada para cada um dos três ativos (ITUB4.SA, PETR4.SA e VALE3.SA), utilizando o conjunto de dados de teste, compreendendo 117 dias de previsão não vistos durante a fase de treinamento.

A avaliação será realizada de forma qualitativa, através da análise visual dos gráficos que comparam os preços reais com os preços previstos, e quantitativamente com a análise dos valores obtidos a partir da Raiz do Erro Quadrático Médio (RMSE).

6.2.1 Ativo ITUB4.SA (Itaú Unibanco)

O modelo conseguiu prever o comportamento dos preços da ação do Itaú Unibanco, como mostra a figura 6.1, especialmente para $t = 10$ dias em diante.

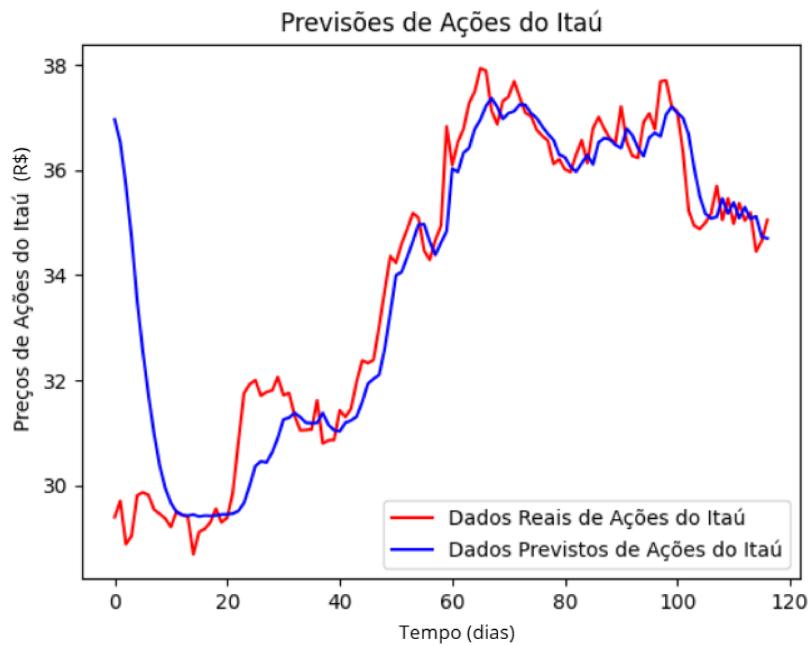


Figura 6.1 – Previsão do modelo (azul) vs. Preço real (vermelho) para ITUB4. Note a alta aderência da previsão ao seguir as flutuações do ativo.

A análise visual do gráfico da figura 6.1 demonstra que o modelo LSTM conseguiu capturar não apenas a tendência geral dos movimentos de preço, mas também grande parte das flutuações de curto prazo, acompanhando os picos e vales com boa precisão. Para o RMSE, obtivemos um valor calculado de 1.4806, o que indica que o preço de fechamento ajustado da ação variou em R\$ 1,48 para mais ou para menos, dentro do intervalo analisado. Portanto, desde que o valor de cada ação do Itaú gire em torno de R\$ 38,00, o erro médio de 4% nos sugere que as características estatísticas intrínsecas da série temporal de ITUB4.SA foram de fato aprendidas pela arquitetura de rede proposta.

6.2.2 Ativo PETR4.SA (Petrobras)

Para as ações da Petrobras o modelo também conseguiu prever a direção geral dos preços, como mostra a figura 6.2.

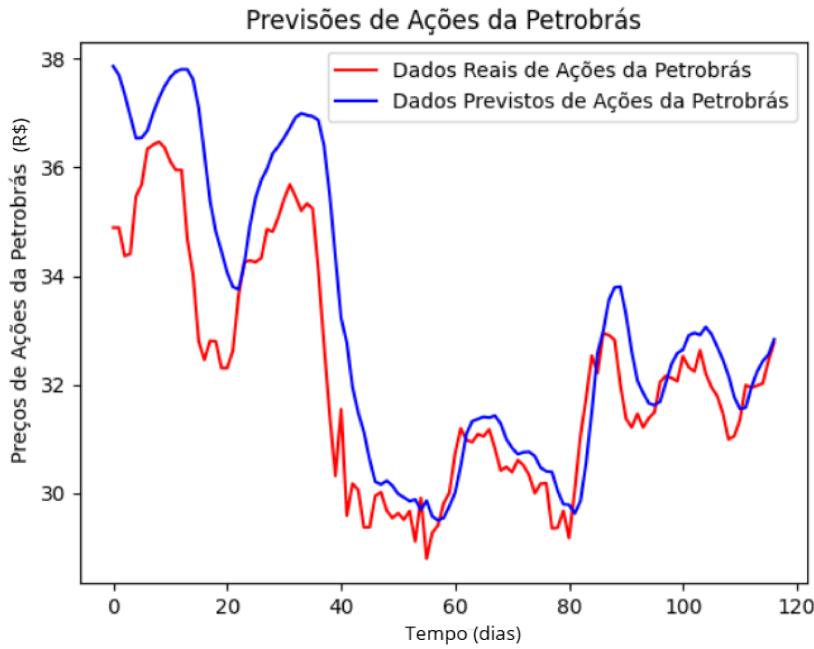


Figura 6.2 – Comparativo entre o preço real da PETR4 e as previsões do modelo. A previsão acompanha a tendência, mas suaviza os picos e vales.

A análise visual do gráfico da figura 6.2 demonstra que o modelo LSTM identificou corretamente as principais tendências de queda e de alta. Entretanto, uma análise mais detalhada revela um atraso temporal (lag) perceptível no início da previsão e um efeito de suavização (smoothing), tornando a previsão menos volátil que a série real. O RMSE para este ativo foi de 1.4650. Como o valor da ação da Petrobrás gira em torno de R\$ 31,00, o erro médio gira em torno de 4,7 %, portanto ligeiramente maior que no caso do Itaú. Ainda, assim, pode-se afirmar que, mesmo para PETR4.SA, o modelo funcionou como um bom indicador de tendência, embora menos preciso.

6.2.3 Ativo VALE3.SA(VALE): Análise de uma Falha Preditiva

A previsão para as ações da Vale S.A. (VALE3.SA) demonstrou uma falha significativa do modelo, como ilustra a figura 6.3.

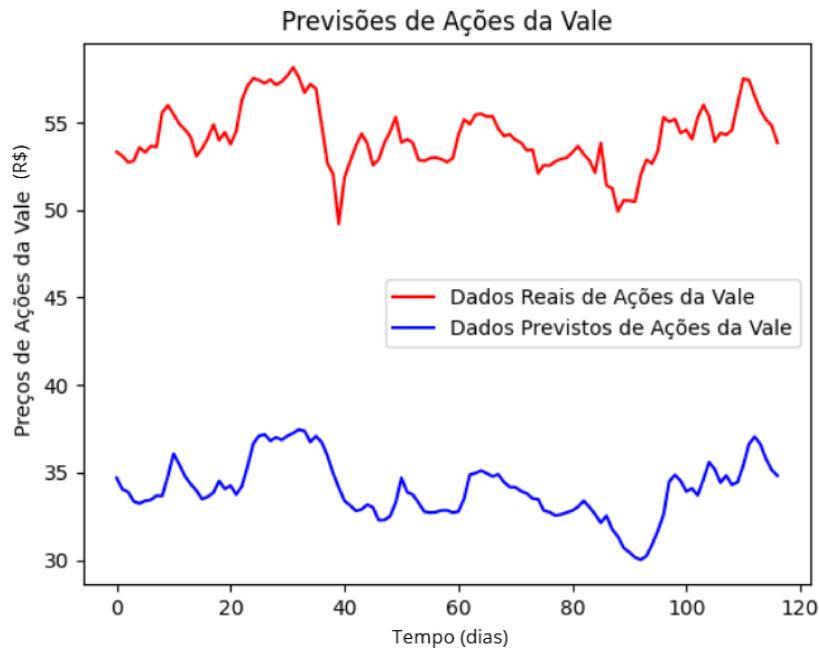


Figura 6.3 – Comparativo entre o preço real e a previsão para VALE3 em dias. O deslocamento vertical evidencia a falha do modelo em prever os preços.

O gráfico exibe um deslocamento significativo (offset) entre o patamar de preços reais e o previsto pelo modelo. Esta discrepância é confirmada quantitativamente pelo alto valor da Raiz do Erro Quadrático Médio (RMSE), que para este ativo atingiu 20,1676. Isso representa 36% do valor atual da ação da Vale (que gira em torno de R\$ 56,00). Para investigar as causas desta falha, foi realizada uma análise aprofundada dos retornos logarítmicos do ativo, comparando-o com os demais. Uma inspeção detalhada do gráfico de sobreposição dos retornos figura 6.4 revela um comportamento particular no período final da série, a partir de 2024.

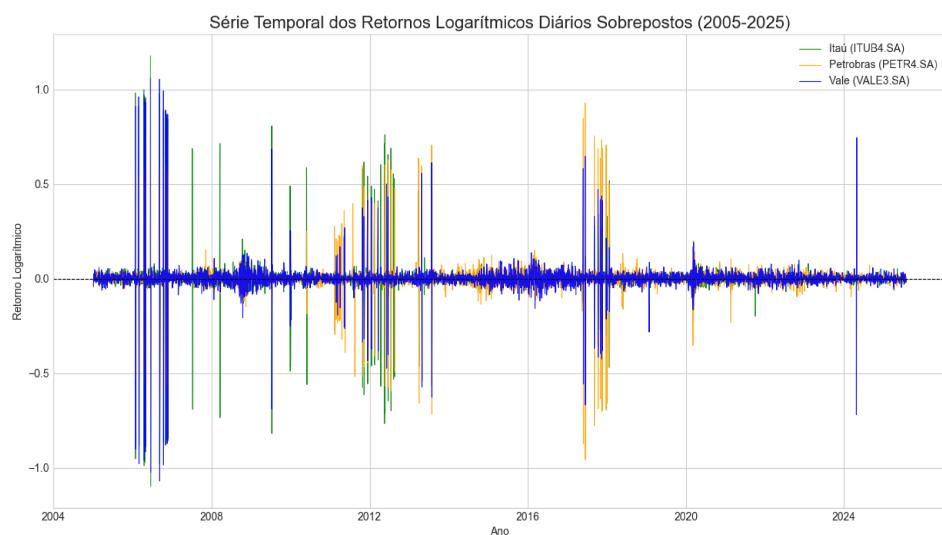


Figura 6.4 – Série Temporal dos Retornos Logarítmicos Diários Sobrepostos (2005-2025)

Nota-se que a série da Vale apresenta um “cluster” de alta volatilidade nesse período específico, o qual não é observado para os outros dois ativos. Nesse período em particular, ocorreu uma grande variação de preços para Vale3.SA, jogando-os para baixo. Como esse período está dentro da parte final de nosso conjunto de treinamento, ele acabou tendo um peso maior em nossos dados de treinamento, o que explica o fato de que nosso modelo, em particular para Vale, previu preços mais baixos, melhor dizendo, bem mais baixos que os reais.

Ocorre que os preços da Vale se recuperaram, mas como nosso modelo prevê preços considerando os preços dos 60 dias de negociação anteriores, o que é um intervalo grande, isso acaba tornando a “recuperação mais lenta”. Em outras palavras, momentos de forte volatilidade podem influenciar os preços previstos por um longo tempo, especialmente quando elaboramos um modelo que considera preços de ativos financeiros como funções de preços históricos do ativo sobre um longo período. Todavia, o comportamento dos preços da Vale ainda assim foi parcialmente capturado pelo nosso modelo.

Perceba que os preços reais da Vale apresentaram um comportamento lateralizado. Além disso um erro sistemático cometido pelo nosso modelo foi verificado, de modo que, está refletido na distância mais ou menos constante entre os níveis de preço real e pređito, cuja origem se apresenta do fato de que os preços desse ativo foram alvo de alta volatilidade em um passado não tão remoto.

O modelo mostrou-se vulnerável a quebras estruturais(volatilidades atípicas), alguns fatores evidenciam essa vulnerabilidade:

- **Fatores Externos e Limitação do Modelo:** O ativo, por ser uma commodity, é fortemente influenciado por fatores externos (ex: preço do minério de ferro, demanda internacional, eventos de grande impacto). Pelo fato do modelo utilizado ser univariado, não foi possível capturar essa complexidade externa, tornando-se inadequado para um cenário de volatilidade atípica.
- **Aleatoriedade do Treinamento:** O processo de treinamento de redes neurais possui componentes aleatórios (inicialização dos pesos e camadas de Dropout). Essa aleatoriedade pode ter levado o algoritmo de otimização a convergir para um "mínimo local" de erro desfavorável, o que resultou em um modelo particularmente sensível à mudança de regime dos dados.

Embora o resultado para a Vale não tenha sido adequado em termos de acurácia, ainda sim foi muito significativo, pois, mostrou empiricamente que uma arquitetura de rede neural pode apresentar uma performance variada dependendo do ativo financeiro, logo, seu comportamento não é universal e um comportamento estacionário da série é um fator crucial para o sucesso da previsão.

CONCLUSÃO

Este trabalho avaliou a eficácia de uma rede neural LSTM na previsão de ativos da bolsa brasileira, obtendo um desempenho variado que revelou tanto o potencial quanto as limitações da abordagem. O modelo conseguiu prever o comportamento geral de preços para os ativos ITUB4.SA e PETR4.SA, mostrando que a arquitetura utilizada tem capacidade para aprender dependências temporais complexas, particularmente para os preços destes ativos. Contudo, a falha significativa na previsão dos preços da VALE3.SA foi a contribuição mais valiosa do estudo, pois mostrou que a metodologia precisa ser aperfeiçoada para aprender a prever o movimento de preços de ativos financeiros quando clusteres de volatilidade estão presentes mais ao final da série temporal.

A análise dos retornos logarítmicos e a natureza do erro sugerem que a discrepância foi causada por uma quebra estrutural na série temporal durante o período de teste. O resultado obtido expõe a principal limitação de um modelo univariado que é: sua incapacidade de se adaptar a novos regimes de volatilidade impulsionados por fatores externos. Conclui-se, portanto, que as redes LSTM são ferramentas poderosas, mas não universalmente eficazes, sua performance depende da estabilidade estatística de cada ativo. Além disso, o estudo corrobora a visão da econofísica de que o mercado financeiro é um sistema complexo que desafia soluções generalistas. A recomendação para trabalhos futuros é a evolução para modelos multivariados, incorporando variáveis externas para criar previsões mais robustas e adaptáveis às mudanças de regime do mercado, a otimização de hiperparâmetros da rede, uma inserção da característica de “humor” do mercado e compreender melhor como clusteres de volatilidade influenciam o comportamento de séries temporais de preços. Esperamos que, a partir daí, tenhamos em mãos elementos que nos permitam elaborar um modelo aperfeiçoado que consiga capturar melhor tais flutuações diárias de preços.

REFERÊNCIAS

- BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE Transactions on Neural Networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.
- BOX, G. E. P.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time Series Analysis: Forecasting and Control**. 5. ed. Hoboken: John Wiley & Sons, 2015.
- BROCKWELL, P. J.; DAVIS, R. A. **Introduction to Time Series and Forecasting**. 3. ed. New York: Springer, 2016.
- CHATFIELD, C. **The Analysis of Time Series: An Introduction**. 6. ed. Boca Raton: Chapman and Hall/CRC, 2003.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.
- HOPFIELD, J. J. Neural network and physical systems with emergent collective computational abilities. **Proceedings of the National Academy of Sciences of the USA**, v. 79, n. 8, p. 2554–2558, 1982.
- HOPFIELD, J. J.; TANK, D. W. “neural” computation of decisions in optimization problems. **Biological Cybernetics**, Springer, v. 52, n. 3, p. 141–152, 1985.
- HOQUE, A. S. M.; ALJAMAAN, A. Can recurrent neural networks predict inflation in euro zone as good as professional forecasters? **Mathematics**, v. 9, n. 19, p. 2486, 2021. Disponível em: <<https://doi.org/10.3390/math9192486>>. Acesso em: 15 ago. 2025.
- HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: Principles and Practice**. 2. ed. Melbourne: OTexts, 2018. Disponível em: <<https://otexts.com/fpp2/>>. Acesso em: 15 ago. 2025.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943.
- MORETTIN, P. A.; TOLOI, C. M. C. **Análise de Séries Temporais: Modelos Lineares Univariados**. São Paulo: Edgard Blücher, 2006. v. 1.
- OLAH, C. **Understanding LSTM Networks**. 2015. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso em: 15 ago. 2025.
- SILVA, E. C.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. 2. ed. São Paulo: LTC, 2016.