**Análise de redes neurais a carros autônomos e ao mercado financeiro**

**Gabriel Duarte Batista de Nazaré¹, Samuel Moraes Batistela².**

1Universidade do Vale do Paraíba/Instituto de Pesquisa e Desenvolvimento, Avenida Shishima Hifumi, 2911, Urbanova - 12244-000 - São José dos Campos-SP, Brasil, [gabrielduartebn@gmail.com](mailto:gabrielduartebn@gmail.com).

2Universidade do Vale do Paraíba/Instituto de Pesquisa e Desenvolvimento, Avenida Shishima Hifumi, 2911, Urbanova - 12244-000 - São José dos Campos-SP, Brasil, samuel.batistela12@gmail.com.

**Resumo**

As redes neurais, inspiradas no funcionamento do cérebro humano, são capazes de reconhecer padrões e resolver problemas complexos. A metodologia aborda as arquiteturas, paradigmas de aprendizagem e funções de ativação das redes neurais. Em veículos autônomos, redes neurais recorrentes são utilizadas para processamento de dados sensoriais, promovendo a navegação segura. No mercado financeiro, uma combinação de redes neurais convolucionais e recorrentes com algoritmos meta-heurísticos otimiza previsões de preços, alcançando uma alta taxa de acerto.

**Palavras-chave**: Redes Neurais, Veículos Autônomos, Mercado Financeiro

**Área do Conhecimento:** Engenharias – Engenharia e Ciência da Computação

**Introdução**

Nos últimos anos, o campo da inteligência artificial (IA) tem experimentado avanços significativos, principalmente devido ao desenvolvimento e aplicação de redes neurais artificiais. As redes neurais, inspiradas na estrutura e no funcionamento do cérebro humano, são sistemas computacionais projetados para reconhecer padrões e resolver problemas complexos de forma eficiente.

O começo do estudo sobre redes neurais se deu em 1943 por Warren McCulloch, um neurofisiologista e psiquiatra americano que estudou na Universidade de Yale, e Walter Pitts, um lógico autodidata. Juntos, no trabalho publicado com o título "*A Logical Calculus of the Ideas Immanent in Nervous Activity*", propuseram um modelo matemático do neurônio biológico baseado em lógica limiar, onde poderia realizar operações lógicas básicas [1]. Este modelo, conhecido como neurônio MCP (McCulloch-Pitts), estabeleceu a base para as futuras implementações computacionais das redes neurais​​ [2].

No ano de 1949 um psicólogo canadense chamado Donald Hebb publicou um estudo chamado “*The Organization of Behavior*”. Este artigo explora o funcionamento do cérebro humano, introduzindo o conceito de que os caminhos neurais são fortalecidos a cada vez que são utilizados, fundamental para o processor de aprendizado e adaptação [3]. Posteriormente o mundo de estudos de inteligências artificiais adotou esse pensamento, a chamando de regra de Hebb.

Chegando em 1958, Frank Rosenblatt, psicólogo americano considerado por alguns como o pai do *deep learnig,* começou a desenvolver e implementar o primeiro modelo de rede neural, chamado de percéptron. Este modelo foi constituído por neurônios MPL (Percéptrons de Múltiplas Camadas), formando uma rede neural simples de apenas uma camada de entrada e uma camada de saída [3].

Isto possibilitou que inúmeros modelos pudessem ser criados desde a sua concepção, essas redes têm se mostrado ferramentas poderosas para uma variedade de aplicações, incluindo reconhecimento de fala, visão computacional, tradução automática, diagnósticos médicos e muitas outras áreas.

Neste trabalho iremos abordar alguns dos conceitos-chave, como a arquitetura, os paradigmas de aprendizagem e as funções de ativação. Além disso, serão analisadas duas aplicações de uma rede neural no mundo real, uma sendo na confecção de carros autônomos e quando aplicado ao mercado financeiro.

**Conceitos básicos**

**1. Arquitetura**

As redes neurais possuem diferentes arquiteturas, e a definição dessa arquitetura é essencial para lidar com diferentes tipos de dados e resolver problemas. A arquitetura de uma rede neural define o tipo de alimentação da informação e como esses dados são processados pelo sistema.

Uma rede de propagação para frente, ou rede de alimentação direta, é caracterizada pelo fluxo unidirecional de informações entre as suas camadas. Também chamadas de nós, as camadas em uma rede de propagação para frente normalmente são dispostas em camada de entrada, camadas intermediárias ou ocultas, e camada de saída.

Figura 1 - Rede de propagação para frente

Diagrama

Descrição gerada automaticamente

Fonte: Site Deep Learning Book

A Figura 1 demonstra o funcionamento de uma rede feedforward. A entrada de uma informação é feita pelos nós de entrada, depois é alimentada para os nós ocultos e, em seguida, os dados são enviados para os nós de saída, mostrando os resultados obtidos pelo processamento da rede neural.

Outro tipo de arquitetura é a que possibilita a realimentação da informação: a rede recorrente. Esta rede permite que os dados fluam em duas direções e que exista uma retroalimentação, permitindo a manutenção de uma 'memória' de informações.

Figura 2 - Rede realimentada

Diagrama

Descrição gerada automaticamente

Fonte: Site Deep Learning Book

A Figura 2 mostra como o processo de realimentação funciona em uma rede recorrente, onde existe uma realimentação de dados nos nós ocultos e uma realimentação dos dados da camada de saída.

A diferença nas tratativas dos dados nessas arquiteturas molda o modelo de rede neural para ser mais adequado ou otimizado para certos tipos de problemas, quando comparado a outros diversos tipos de arquiteturas. Um exemplo disso é a classificação e o reconhecimento de padrões, em que uma rede *feedforward* é mais utilizada para resolver esse problema em comparação a uma rede recorrente. Por sua vez, a rede recorrente é mais utilizada em sequências temporais e contextos de longo prazo, como tradução automática.

**2. Paradigmas de aprendizagem**

Os paradigmas de aprendizagem se referem a diferentes abordagens e metodologias utilizadas para treinar redes neurais. Dois dos paradigmas mais relevantes são a aprendizagem supervisionada e a aprendizagem não supervisionada. Cada um deles oferece abordagens distintas para resolver problemas com base nos dados disponíveis e nos objetivos do treinamento.

A aprendizagem supervisionada é um modelo no qual a rede neural é treinada utilizando dados rotulados, sendo cada exemplo no conjunto de treinamento composto por uma entrada e a saída desejada correspondente. Neste tipo de modelo, a importância está em fazer o mapeamento correto das saídas para as respectivas entradas, priorizando a precisão nas previsões futuras quando testado com novos dados. Isso é feito ao comparar as saídas de dados da rede com as saídas desejadas, permitindo ajustar os pesos para direcionar a saída para um valor desejado para a resolução do problema. Em um modelo matemático, isso se assemelha a uma regressão linear, onde existe um parâmetro e os dados que estão mais discrepantes serão descartados.

Figura 3 - Regressão linear

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Fonte: Site Dicionário financeiro

Já a aprendizagem não supervisionada, diferentemente da supervisionada, não atua com dados rotulados. O objetivo dessa rede não é a predição de uma saída específica, mas identificar padrões e atributos estatisticamente relevantes nos dados, isso permite que a análise de uma alta quantidade de dados seja realizada de uma forma mais exploratória para possibilitar uma aglomeração de padrões.

**3. Função de ativação**

As funções são um elemento importante para a elaboração de redes neurais, definindo se um nó deve ser ativado ou não, com base na relevância da informação para o problema em questão. Isso possibilita que certas informações possam ser descartadas, introduzindo uma não-linearidade nos modelos e permitindo que a rede aprenda a fazer representações complexas de dados [4]. A seguir serão comentados alguns dos modelos mais utilizados de funções de ativação.

**3.1 Função de escada – Função binária**

Figura 4 - Gráfico da função binaria

Figura 5 - Representação binaria

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixaTexto, Carta

Descrição gerada automaticamente

Fonte: Site v7labs

Este é um modelo simples, restringindo as saídas das redes neurais em valores binários [0, 1]. Esta saída tem como influência a superação de um valor predeterminado, definido como um limiar. Ao receber uma informação que atinja ou passar esse limiar, será retornado 1 e 0 caso contrário. Essa função é normalmente utilizada em perceptrons simples.

**3.2 Função Linear**

Figura 6 - Gráfico da função linear

Figura 7 - Representação linear

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Texto, Carta

Descrição gerada automaticamente

Fonte: Site v7labs

Este modelo também é um dos modelos mais simples de funções de ativação, onde a saída é proporcional a sua entrada. Ao falar em termos matemáticos, pode-se dizer que ao utilizar funções lineares nas camadas de uma rede neural, a rede inteira seria resumida em uma combinação linear de entradas. Assim como o modelo passado, esse modelo limita a capacidade de aprendizado de padrões complexos presentes nos dados.

**3.3. Função sigmoidal**

Figura 8 - Gráfico da função sigmoide

Figura 9 - Representação Sigmoide

Diagrama, Esquemático

Descrição gerada automaticamenteTela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Fonte: Site v7labs

Esta função opera de forma semelhante a uma função binária, mas com a diferença na maneira como ocorre a mudança de estados. Em vez de uma transição abrupta de 0 para 1, a transição é suavizada. A aplicação desse modelo resulta em uma saída não linear, o que permite à rede aprender limites de decisão mais complexos e assumir valores intermediários entre 0 e 1.

**3.4 Tangente hiperbólica (tanh)**

Figura 11 - Gráfico da função de Tanh

Figura 10 - Representação Tanh

Uma imagem contendo Gráfico

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Fonte: Site v7labs

Esta função é similar à função anterior, mas possui a capacidade de mapear valores reais para o intervalo [-1, 1], o que pode levar a uma maior eficiência na aprendizagem, pois a média das ativações ao longo das camadas pode ser mais próxima de zero, ajudando a manter os gradientes em uma faixa útil.

**4. Adaptação de Pesos**

A adaptação de pesos em redes neurais é um processo essencial para o aprendizado, onde os pesos das conexões entre neurônios são ajustados durante o treinamento para minimizar o erro entre a saída prevista pela rede e a saída desejada. Este processo é geralmente realizado utilizando métodos de otimização baseados em gradiente [5]. Abaixo iremos comentar sobre algumas das regras de adaptação de pesos.

**4.1 Regra de Hebb**

A regra de Hebb baseia-se na ideia de que a simultaneidade da atividade neuronal fortalece as sinapses. Quando um neurônio pré-sináptico A frequentemente contribui para disparar um neurônio pós-sináptico B, a conexão entre esses neurônios é reforçada [5]. A alteração no peso sináptico Δw entre dois neurônios é proporcional ao produto das suas ativações:

Δwij = η.xi.​yj​

A variável n nesta função determina a velocidade de aprendizagem. Esta regra foi adaptada para a sua utilização em redes neurais artificiais para que seja possível ter um aprendizado associativo, onde as conexões entre neurônios são ajustadas com base nas correlações de suas atividades, permitindo que a rede lembre padrões associados.

**4.2 Regra de Delta**

A regra delta busca minimizar a diferença entre a saída desejada e a saída real do neurônio ajustando os pesos sinápticos. A atualização dos pesos é feita de maneira iterativa com base no erro observado. O objetivo é encontrar os pesos que melhor aproximem as previsões do modelo às verdadeiras respostas. A fórmula utilizada para a atualização dos pesos é:

Δwi​=η.(d−y).xi

A diferença nessa fórmula é a introdução de um objetivo ou valor desejado, este é representado pela variável d. A diferença (d – y) representa o erro entre a saída desejada e a saída atual do neurônio. Esta regra é mais utilizada em aplicações lineares devido a sua taxa de aprendizado, que se torna sensível a valores muito altos e a valores muito baixos, podendo levar a oscilações e a convergências lentas, respectivamente.

**4.3 Aprendizagem competitiva**

A aprendizagem competitiva é uma técnica utilizada em redes neurais artificiais, especialmente em redes não supervisionadas, onde os neurônios competem entre si para serem ativados. Esse tipo de aprendizagem é fundamental em tarefas como a clusterização e a auto-organização de dados. Em cada iteração, um conjunto de neurônios compete para responder a um conjunto de entrada. Apenas o neurônio vencedor, geralmente o mais próximo da entrada em termos de alguma medida de similaridade (como a distância euclidiana), é atualizado. Para esse modelo, é utilizado uma regra de Hebb modificada, descrita abaixo:

Δwi​ = η.(x−wi​)

Esta regra serve para que a rede se auto-organize de forma que diferentes neurônios respondem a diferentes padrões ou clusters dos dados de entrada, tendo uma redução na dimensionalidade dos dados e gerando uma eficiência computacional.

**Aplicações**

**1. Carros autônomos**

Em carros autônomos, as redes neurais são treinadas para reconhecer padrões e tomar decisões com base em dados sensoriais recebidos de câmeras, radares, LiDARs e outros sensores.

Nessa etapa, iremos analisar a utilização de redes neurais do artigo "Learning a Driving Simulator" realizado pela empresa Comma.ai [6]. O produto dessa empresa é a transformação de carros comuns em carros autônomos, fazendo a integração de carros tradicionais a câmeras, sensores e um software embarcado que contém a inteligência artificial que irá fazer a automatização.

O artigo explora uma abordagem para criar um simulador de direção que possa prever sequências de vídeo realistas de estradas usando técnicas de aprendizado de máquina. O objetivo é desenvolver um agente que possa aprender a simular eventos futuros na estrada e ajudar no planejamento de manobras para carros autônomos. A abordagem se baseia em variações de autoencoders e redes neurais recorrentes condicionadas por ações.

Autoencoder é um tipo de rede neural usada principalmente para aprendizado não supervisionado, com o objetivo de aprender a codificar dados de entrada em uma representação de menor dimensão e depois reconstruí-los de volta para algo semelhante ao original.

O primeiro passo é usar um autoencoder para reduzir a dimensionalidade das imagens de vídeo capturadas pela câmera do carro. Isso envolve converter as imagens em uma representação compacta em um espaço latente, resultando em imagens que mantêm detalhes importantes, como marcas na estrada e bordas de carros, sendo esses detalhes utilizados no processamento da rede neural.

Com as informações fornecidas do treinamento do autoencoder, a rede neural irá atuar no espaço latente criado utilizando uma rede neural recorrente para responder a ações específicas do carro. possibilitando a previsão de estados futuros no espaço latente para conseguir prever as ruas e as aproximações de carros.

Também é utilizado a *Backpropagation* para reajuste de pesos e vieses das conexões entre os neurônios, suavizando a diferença entre a saída esperada e a saída real da rede.

A utilização de uma rede neural recorrente se demonstra adequada para os objetivos desse projeto. Devido a arquitetura da rede ser composta pela realimentação de dados, isso ajuda a previsão dos passos futuros por conta da sua capacidade de manter informações sobre os estados anteriores, capturando dependências temporais e a dinâmica do ambiente. A aplicação dessas técnicas pode transformar carros comuns em veículos autônomos, promovendo avanços significativos na segurança e eficiência dos sistemas de direção autônoma.

**2. Previsão de Mercados Financeiros**

No artigo "Hybrid Neural Network-Based Metaheuristics for Prediction of Financial Markets: A Case Study on Global Gold Market" [7], as redes neurais são aplicadas para prever variações de preços em mercados financeiros, especificamente o mercado global de ouro. A pesquisa foca no desenvolvimento de métodos híbridos que combinam redes neurais com algoritmos meta-heurísticos para melhorar a precisão das previsões financeiras.

A abordagem proposta utiliza uma combinação de redes neurais convolucionais (CNN) e unidades recorrentes bidirecionais com portas (BiGRU), cuja otimização dos hiperparâmetros é feita pelo algoritmo meta-heurístico Firefly. Adicionalmente, o algoritmo de otimização Moth-Flame é empregado para selecionar as variáveis mais influentes sobre a variável alvo. Esta combinação de técnicas visa identificar e melhorar os sinais provenientes de indicadores de análise técnica, que tradicionalmente apresentam grandes margens de erro.

O principal objetivo é desenvolver um modelo capaz de prever com alta precisão os sinais de compra e venda no mercado de metais preciosos. A utilização de meta-heurísticas para otimização dos modelos de deep learning permite uma análise mais precisa e robusta dos dados de mercado, possibilitando um suporte decisório eficaz para investidores enfrentarem as incertezas dos mercados financeiros.

Os resultados indicam que o modelo proposto tem uma taxa de acerto em torno de 94%, demonstrando uma capacidade superior de previsão em comparação com outros métodos de aprendizado de máquina e deep learning presentes na literatura. Essa abordagem inovadora pode ser utilizada como uma ferramenta de suporte à decisão, ajudando investidores a mitigar riscos e a maximizar lucros em mercados voláteis como o de metais preciosos.

**Conclusão**

As análises das aplicações em carros autônomos e previsão de mercados financeiros demonstram o poder e a versatilidade das redes neurais em diferentes contextos. Nos carros autônomos, as redes neurais recorrentes e autoencoders permitem uma navegação segura e preditiva, enquanto na previsão de mercados financeiros, a combinação de técnicas de deep learning com meta-heurísticas proporciona uma ferramenta robusta para análise e decisão. Essas aplicações evidenciam o potencial transformador das redes neurais na resolução de problemas complexos, aprimorando a tomada de decisão e a automação em diversos setores.

**Referências**

[4] BAHETI, P. Activation functions in neural networks [12 types & use cases]. V7labs.comV7,, 10 abr. 2024. Disponível em: <https://www.v7labs.com/blog/neural-networks-activation-functions>. Acesso em: 3 jun. 2024

[3] CARDON, André; MÜLLER, Daniel Nehme; NAVAUX, Philippe. **Introdução às redes neurais artificiais**. Porto Alegre, 1994.

[2] FLECK, Leandro et al. Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47-57, 2016.

[1] PINHEIRO, A. Redes neurais artificiais. Disponível em: <https://medium.com/@augusto_Pinheiro/redes-neurais-artificiais-133de77c7240>. Acesso em: 31 maio 2024.

[5] RAUBER, Thomas Walter. **Redes neurais artificiais**. Universidade Federal do Espírito Santo, v. 29, 2005.

[6] SANTANA, E.; HOTZ, G. **Learning a Driving Simulator**. 2016. Disponível em: <http://arxiv.org/abs/1608.01230>. Acesso em: 6 jun. 2024.

[7] MOUSAPOUR MAMOUDAN, M. et al. Hybrid neural network-based metaheuristics for prediction of financial markets: a case study on global gold market. **Journal of computational design and engineering**, v. 10, n. 3, p. 1110–1125, 2023.