INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo

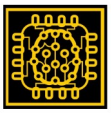
EAILab
Laboratório de Inteligência Artificial Embarcada
Instituto Federal de São Paulo

COLEÇÃO DE POSTS EAILAB

2023-2024

Volume 1
2024

Sumário



EAILab
Laboratório de Inteligência Artificial Embarcada
Instituto Federal de São Paulo

Introdução

Este documento reúne uma coletânea dos *posts* relacionados a inteligência artificial (IA), aprendizado de máquina (*machine learning* – ML), Tiny ML e computação na borda (*edge computing*) publicados na página do EAILab no período entre 2023 e 2024.

O objetivo deste documento é servir como uma referência preliminar sobre assuntos relacionados à IA, em especial à IA embarcada, diferentes estratégias de aplicação, tendências e evolução.

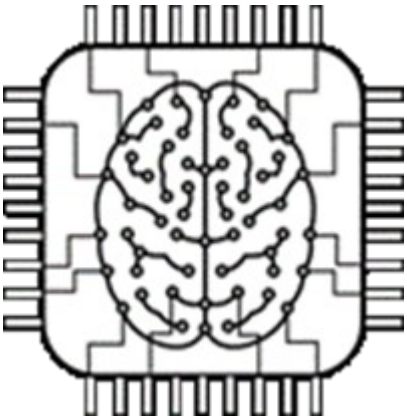
Dr. Arnaldo de Carvalho Junior
EAILAB – IFSP
https://eailab.labmax.org/
Volume 1
Versão 1
Novembro 2024.

Post 1: 10 Tendências de Aplicação de Inteligência Artificial em 2024!

Há um século, a economia dependia fortemente de trabalho manual. Com o acelerado desenvolvimento da eletrônica e informática, principalmente após a segunda guerra mundial, várias tecnologias capazes de realizar cálculos complexos foram introduzidas. Assim, em poucas décadas a humanidade saiu dos processos manuais de cálculo como ábaco para os mais recentes computadores quânticos. Essas tecnologias encontraram ampla aplicação em uma variedade de indústrias, incluindo pesquisa e desenvolvimento, defesa, medicina, processos industriais, entre outros.

Já em 1956, a inteligência artificial (IA) ficou em evidência, através do artigo "*Computing Machinery and Intelligence*" publicado pelo matemático inglês Alan Turing. Entretanto, os avanços recentes de hardware e software permitiram que a inteligência artificial ganhasse meios e massa crítica para se estabelecer como ciência integral, multidisciplinar, com aplicações nas mais diversas áreas do conhecimento humano.

A seguir são relacionadas 10 tecnologias com IA em destaque em 2024:



1. **Plataforma de Aprendizado de Máquina (*Machine Learning* – ML):** tem o objetivo de desenvolver novos métodos para que os computadores aprendam e se tornem mais inteligentes. Algoritmos, interfaces de programação de aplicação (*Application Programming Interfaces* – APIs), desenvolvimento, ferramentas de treinamento, dados massivos e aplicações estão impulsionando o uso de ML em classificação e previsão.

2. **Plataformas de aprendizagem profunda (*Deep Learning* – DL):** tipo de ML que analisa dados e descobre tendências usando circuitos neurais semelhantes aos encontrados no cérebro humano. Exemplos de aplicações são reconhecimento de tela automatizado, reconhecimento de imagem e previsão de qualquer coisa detectável no domínio digital.
3. **Reconhecimento de Fala (*Speech Recognition*):** método de converter a fala humana de modo que o software do computador possa entender. Traduzir e transformar a linguagem humana em formatos úteis está se expandindo rapidamente, inclusive para ações inclusivas. Diversas Corporações fornecem serviços de reconhecimento de fala.
4. **Geração de linguagem natural:** ramo da tecnologia IA que converte o texto em dados e auxilia os sintomas a transmitir perfeitamente suas idéias e pensamentos, facilitando a comunicação entre humanos de diferentes culturas e na interação com as máquinas. Destaque para os Modelos de Linguagem de Grande Escala (*Large Language Models* – LLMs), que são modelos de ML utilizando DL para processar e entender a linguagem natural. Os LLMs são treinados em grandes volumes de dados da internet, aprendendo padrões sobre como as palavras e frases são normalmente usadas em conjunto. Quando carregado com uma nova entrada de texto, um LLM tentará gerar a continuação mais provável desse texto com base no que aprendeu durante o treinamento. Exemplo de LLM com destaque na mídia é o ChatGPT.
5. **Agentes Virtuais:** softwares que permitem a máquina se comunicar e interagir efetivamente com os humanos. A IA está sendo cada vez mais empregada no suporte ao cliente por meio de Chatbots e como um gerenciador de casa inteligente.
6. **Gerenciamento de Decisão (*Decision Management*):** Pode-se incorporar IA em robôs preparando-os para treinamento, gerenciamento e ajuste. As organizações já estão incorporando o gerenciamento de decisões em seus aplicativos para planejar e executar decisões automáticas que agregam valor ao negócio e aumentam sua viabilidade. Além disso, as técnicas de IA e ML, treinadas e validadas por um especialista humano, podem ser utilizadas em sistemas especialistas para tomada de decisão (decision making expert systems) com aplicações em diagnóstico médico, controle automatizado de sistemas, veículos autônomos, entre outras finalidades.
7. **Automação Robótica de Processos:** Automação é a execução de tarefas por técnicas computadorizadas ou mecânicas que outrora era executada por humanos. Ela destina-se a aumentar e complementar as habilidades humanas, bem como executar tarefas repetitivas ou que envolvam algum grau de risco.
8. **Biométrica:** a ciência de reconhecer, medir e analisar as características físicas do corpo humano. Ela permite cria interações orgânicas entre humanos e máquinas, utilizando coisas, imagens, palavras e linguagem corporal. É usado principalmente para pesquisas de mercado.

9. **Hardware otimizado para IA:** dispositivos estão sendo projetados e construídos com processamento e gráficos otimizados para executar tarefas relacionadas à IA. Organizações estão investindo pesadamente em IA para acelerar o desenvolvimento da próxima geração de serviços, como Google, Intel, e Nvidia.
10. **Defesa Cibernética:** mecanismo de defesa do computador que detecta, previne e mitiga ataques e ameaças contra o sistema e a infraestrutura de dados. Pode-se combinar redes neurais artificiais (*Artificial Neural Networks* – ANNs) com técnicas de aprendizagem de máquina para para descobrir a atividade suspeita do usuário e detectar riscos de segurança cibernética.

REFERÊNCIA

N. Duggal. Top 10 Artificial Intelligence Technologies in 2024, SimpliLearn. Published in Nov, 24, 2023. Disponível em: https://www.simplilearn.com/top-artificial-intelligence-technologies-article. Acessado em 28 de novembro de 2023.

Elaborado Por: Dr. Arnaldo de Carvalho Junior.

Publicado em: Nov 28, 2023.

Disponível em: https://eailab.labmax.org/2023/11/28/10-tendencias-de-aplicacao-de-inteligencia-artificial-em-2024/. Acessado em Nov 08, 2024.

Post 2: Tendências em Edge Computing

A internet das coisas (*internet of things* – IoT) refere-se a um massivo conjunto de dispositivos conectados em rede. A internet facilita a comunicação entre os próprios dispositivos, sistemas de gerenciamento e a nuvem (cloud).

O termo IoT surgiu em 1985 [1], antes mesmo de a internet se tornar popular. O termo refere-se à integração entre pessoas, processos e tecnologias via dispositivos conectáveis e acessíveis por qualquer tecnologia.

Entre as principais aplicações de IoT pode-se destacar a Indústria 4.0 ou internet das coisas industriais (*industrial internet of things* – IIoT), logística 4.0, agricultura 4.0, telemedicina, transporte 4.0, entre outros [2].

Figura 1: Aplicações de Smart IoT (adaptado de [2]).

A integração da inteligência artificial (artificial Intelligence – AI), aprendizado de máquina (*machine learning* – ML) e aprendizado profundo (deep learning) ao sistema IoT amplia o leque de possibilidades, aumentando a automatização e o poder de decisão dos sistemas IoT, tornando-os inteligentes (*smart*). A inteligência artificial das coisas (*artificial Intelligence of things* – AIoT) expande as capacidades analíticas e preditivas, amplificando muito o valor dos dados coletados [3].

Figura 2: AIoT.

A quantidade crescente de dispositivos conectados, a velocidade de conexão, a capacidade de processamento dos sistemas na nuvem e a latência podem afetar a performance do sistema IoT ao ponto de ser eficaz em momentos críticos de tomada de decisão. Para esses casos, surge a computação de borda (*edge computing*), onde o processamento ocorre o mais próximo possível da fonte de dados [4]. É o caso, por exemplo, dos veículos conectados e autônomos.

Figura 3: Edge Computing em Veículos Autônomos (Adaptado de [5]).

A seguir são apresentadas as principais tendências em ***Edge Computing*** [7,8]:

11. **Computação em tempo real (*real-time computing*):** a necessidade de respostas imediatas, sem esperar pelo processamento em nuvem, está forçando o desenvolvimento acelerado de soluções de edge computing.
12. **Inteligência Artificial na Borda (*edge artificial Intelligence* – EAI):** Soluções de edge computing, incluindo técnicas de AI e ML em dispositivos compactos (*tiny* ML), capazes de realizar

análises complexas para a tomada de decisões, aumentam a eficiência e reduzem a necessidade de enviar grandes quantidades de dados para a nuvem.

13. **Segurança na borda (*edge security*):** garante a integridade e confidencialidade dos dados, reduzindo a exposição e possíveis ameaças durante a transferência para a nuvem. Assim, soluções de criptografia ponta-a-ponta, autenticação de dispositivos e monitoramento em tempo real são críticos para o edge computing.
14. **Orquestração da borda para a nuvem (*edge-to-cloud orchestration*):** tornar a transição de dados entre a borda e a nuvem tão suave quanto possível proporcionará uma experiência de usuário mais consistente. Isso é possível com a orquestração eficiente entre a borda e a nuvem, garantindo a coesão nos ecossistemas de computação distribuída e a otimização do fluxo de dados através de ferramentas avançadas de gerenciamento do sistema.
15. **Conectividade borda-a-borda (*edge-to-edge connectivity*):** fundamental para uma colaboração eficiente entre dispositivos, em aplicações que exigem ação coordenada em tempo real.
16. **Integração com 5G:** tecnologia de redes de comunicação móvel 5G garante não só maior velocidade de transmissão de dados, mas também redução significativa da latência. A expansão de redes 5G, tanto públicas quanto privadas, facilitará a rápida transferência de dados entre dispositivos e pontos de processamento próximos.
17. **Convergência entre tecnologia da informação e tecnologia da operação (*information technology* (IT) – *operational technology* (OT) *convergence*):** A convergência IT/OT melhora a integração e a colaboração de domínios tradicionalmente separados, permitindo a comunicação consistente e troca de dados entre dispositivos de borda e sistemas backend. Além disso, os sistemas integrados de IT/OT podem erradicar operações desarticuladas e isoladas, estabelecendo coordenação e fluxos de informação eficientes. A convergência IT/OT ainda auxilia no gerenciamento e controle unificados da infraestrutura de IT e OT, levando a uma maior eficiência operacional e otimização de recursos. Análises avançadas, monitoramento em tempo real e manutenção preditiva na borda, melhorando a confiabilidade e o desempenho dos sistemas de computação na borda, são possíveis. A fusão IT/OT promove, assim, protocolos padronizados e interoperabilidade entre dispositivos.
18. **Sustentabilidade na borda (*sustainability at the edge*):** o edge computing pode auxiliar no desenvolvimento de dispositivos e sistemas que otimizem o consumo de energia e minimizem a pegada de carbono, promovendo equilíbrio entre inovação e responsabilidade social.

Figura 4: Edge Computing (Adaptado de [8]).

Com a evolução da tecnologia edge computing haverá uma migração massiva da nuvem para a borda [6]. Processadores pequenos e poderosos, ferramentas matemáticas robustas e algoritmos eficazes usando poucos recursos computacionais serão essenciais em aplicações de *edge computing*.

REFERÊNCIAS

[1] Sharma, C. Correcting the IoT history. 14 mar 2016. Disponível em: https://www.chetansharma.com/correcting-the-iot-history/. Acessado em 12 dez 2023.

[2] Whaiduzzaman, M.; Barros, A.; Chanda, M.; Barman, S.; Sultana, T.; Rahman, M.S.; Roy, S.; Fidge, C. A Review of Emerging Technologies for IoT-Based Smart Cities. Sensors 2022, 22, 9271. https://doi.org/10.3390/s22239271.

[3] Kamal, R. Use Cases And Advantages Of AIoT: Merging AI And IoT technologies. 2022. Disponível em: https://www.intuz.com/blog/use-cases-and-advantages-of-aiot.. Acessado em Dez 12, 2023.

[4] CBInsights. What is Edge Computing? 11 mar 2021. Disponível em: https://www.cbinsights.com/research/what-is-edge-computing/. Acessado em Dez 12, 2023.

[5] Lu, S.; Shi, W. Vehicle Computing: Vision and challenges, Journal of Information and Intelligence, Vol. 1, Issue 1, 2023, pp. 23-35, ISSN 2949-7159, https://doi.org/10.1016/j.jiixd.2022.10.001. Acessado em Dez 12, 2023.

[7] StartUs Insights. Explore the Top 10 Edge Computing Trends in 2024. Disponível em: https://www.startus-insights.com/innovators-guide/edge-computing-trends/. Acessado em Dez 12, 2023.

[8] Kasam, A. Top Five Edge Computing Trends to Watch Out for in 2024. 16 out 2023. Disponível em: https://enterprisetalk.com/featured/top-five-edge-computing-trends-to-watch-out-for-in-2024/. Acessado em Dez 12, 2023.

Elaborado Por: Dr. Arnaldo de Carvalho Junior
Publicado em: Dez 12, 22023
Disponível em: https://eailab.labmax.org/2023/12/12/tendencias-em-edge-computing/. Acessado em Nov 08, 2024.

Post 3: PAL Eт: A Solução para os Desafios de AIoT em Edge Computing

1. INTRODUÇÃO

A inteligência artificial - IA (*artificial intelligence* – AI) tem se desenvolvido de forma acelerada nos últimos anos, com aplicações nas mais diversas áreas do conhecimento humano. Aprendizado de máquina (*machine learning* – ML), reconhecimento de fala, geração de linguagem natural, aprendizagem profunda (*deep learning* – DL), biometria, diagnósticos, cibersegurança, automação e robótica estão entre as principais tendências de desenvolvimento futuros em AI [1].

Outra frente de destaque de desenvolvimento tecnológico deste século é a internet das coisas (*internet of things* – IoT) e seus mais diversos campos de interesse, como indústrias, automação de cidades, agronegócio, logística, infraestrutura, entre outros [2].

A adição da AI ao IoT eleva este último a um novo patamar, aumentando a eficiência das operações de tecnologia da informação (*information technology* – IT), as interações entre homem e máquina, melhor gerenciamento e análise dos dados. A AI + IoT cria um novo paradigma tecnológico, chamado de inteligência artificial das coisas (*artificial intelligence of things* – AIoT) [2].

A arquitetura da IoT pressupõe uma massiva quantidade e diversidade de sensores, conectividade via rede de telecomunicações, gerando um grande volume de dados (big data) que é direcionado para processamento em nuvem ou em camada intermediária da arquitetura (*fog*) [2].

No entanto, há situações em que a análise, diagnóstico e tomada de decisão devem ser realizadas de forma ágil e rápida, sem atrasos. Nesses casos, o processamento ocorrendo o mais próximo da fonte de dados (sensores), o chamado processamento de borda (edge computing), é importante [2]. Infelizmente, os dispositivos de edge computing e sistemas embarcados (*embedded systems*) podem ter recursos computacionais limitados. A implementação de algoritmos de AI, ML, e sistemas para tomada de decisões que apresentem baixa complexidade e consumam poucos recursos são desejáveis. Neste cenário, uma tecnologia que se destaca é a lógica paraconsistente anotada evidencial (*paraconsistent annotated evidential logic* – PAL Eт) [3].

A PAL Eт pertence à família de lógicas paraconsistentes, podendo lidar com informações inconsistentes e contraditórias, sem que o peso dos conflitos invalide as conclusões. A PAL Eт possui algoritmo robusto, matemática de baixa complexidade, reversibilidade, poucas regras e decisões. Por estas características

ela vem ganhando atração entre pesquisadores das mais diversas áreas, de automação e robótica a sistemas de diagnóstico médico e tomada de decisão [3].

2. UM POUCO SOBRE PAL ET

A PAL ET utiliza 1 ou mais variáveis, geralmente um par (μ_1,μ_2), como anotações ou evidências em um diagrama de Lattice de 4 vértices, expressam o conhecimento sobre uma determinada proposição ***P***. O poder de representar evidências e analisar contradições no diagrama de Lattice da PAL ET ficou demonstrado em uma ampla gama de pesquisas, desde partir do cálculo diferencial integral paraconsistente, ao modelo de Bohr para níveis de energia do Hidrogênio átomo, energia escura e teoria da relatividade, e até uma derivação para a lógica paraquântica [3].

Uma das variáveis suporta a proposição ***P***, sendo chamada de grau de evidência favorável ($\mu = \mu_1$). A outra variável será contrária à proposição ***P***, sendo denominada de grau de evidência desfavorável (λ). Se ambas as variáveis suportam a proposição P, utiliza-se o complemento de uma delas como desfavorável ($\lambda = 1 - \mu_2$). A Figura 1 a seguir apresenta o diagrama de Lattice da PAL ET. Uma boa descrição do algoritmo básico (Figura 2) pode ser encontrada na referência [4].

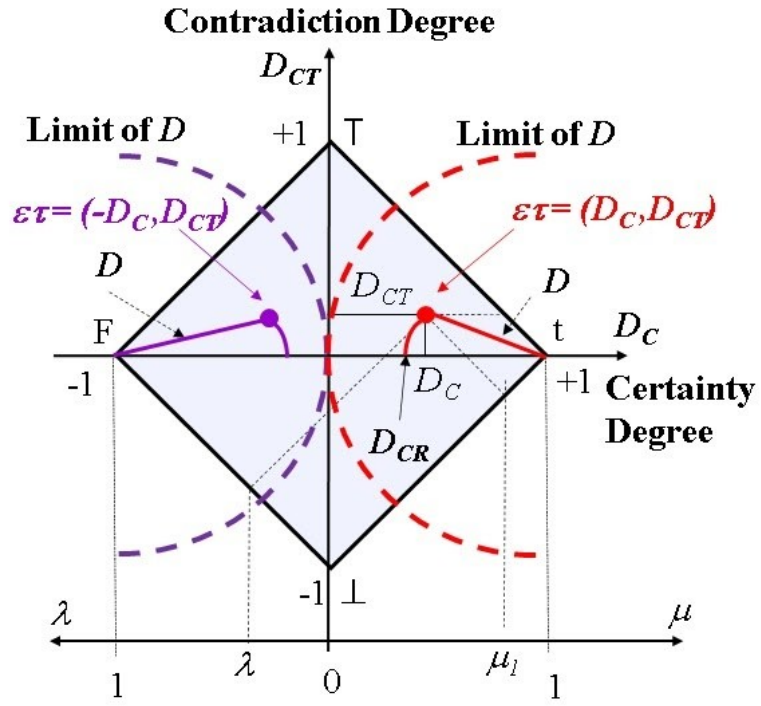


Figura 1 – Diagrama de Lattice da PAL Et [3]

Input values

μ */ favorable evidence degree: $0 \leq \mu \leq 1$

λ */ unfavorable evidence degree: $0 \leq \lambda \leq 1$

Calculate the Contradiction Degree

$$\mu_{ECT} = \frac{D_{CT} + 1}{2} = \frac{\mu + \lambda}{2}$$

Calculate the Certain Interval

$$\varphi_E = 1 - |2\mu_{ECT} - 1|$$

Calculate the Certainty Degree

$$D_C = \mu - \lambda$$

Calculate the Contradiction Degree

$$D_{CT} = \mu + \lambda - 1$$

Calculate the Distance

$$D = \sqrt{(1 - |D_C|)^2 + {D_{CT}}^2}$$

Calcule o Grau de Certeza Real

If $Dc > 0$, $D_{CR} = (1 - D)$

If $Dc < 0$, $D_{CR} = (D - 1)$

Output signal

If $\varphi_E \leq 0{,}25$ or $D > 1$

Let S1 = 0,5 and S2= $\varphi_{E(\pm)}$: Indefinition and go to End

Calculate the Real Certainty Degree

$$\mu_{ER} = \frac{D_{CR}+1}{2}$$

Determine the signal of Certain Interval

If $\mu_{ECT} < 0{,}5$; $\varphi = \varphi_{E(-)}$

If $\mu_{ECT} > 0{,}5$; $\varphi = \varphi_{E(+)}$

If $\mu_{ECT} = 0$; $\varphi = \varphi_{E(0)}$

Output results

Let S1 = μ_{ER} e S2 = $\varphi_{E\ (\pm)}$

End

Figura 2 – Algoritmo básico da PAL Et [4]

3. APLICAÇÕES DA PAL ET EM AIOT E EDGE COMPUTING

A grande vantagem da PAL ET está no uso de regras e operações matemáticas simples, o que permite criar sistemas robustos e confiáveis, mesmo em dispositivos com limitações de recursos computacionais, sendo ideal para aplicações embarcadas e de *Edge Computing*.

3.1 PAL ET ou Fuzzy?

Os computadores têm dificuldade em imitar o raciocínio humano para tomada de decisão quando o valor está entre verdadeiro e falso. Em tal situação, várias técnicas têm sido exploradas, como a lógica difusa (Fuzzy *logic*) [3].

Apesar de amplamente utilizada, a lógica Fuzzy tem algumas limitações, pois pode exigir vários parâmetros, dezenas ou centenas de regras do tipo if-them, funções de associação e inferências. Além disso, o desenho de regras Fuzzy depende do conhecimento e da experiência humana. Se alta velocidade e extensas regras Fuzzy forem requeridas, pode não ser viável utilizá-la em um sistema especialista embarcado, com recursos computacionais limitados. A PAL Eᴛ, por outro lado, pode fornecer velocidade computacional em sistemas especialistas. Simples equações são extraídas das análises no diagrama de Lattice da PAL Eᴛ, facilitando o desenvolvimento de sistemas especializados com incertezas [3].

O diagrama de Lattice da PAL Eᴛ pode ser dividido em um certo número de regiões, ou "estados lógicos", usados para tomada de decisões. Um exemplo típico é a divisão em 12 regiões, utilizadas, por exemplo, para a toma de decisão de direção de um robô móvel, como proposto em [5]. Um exemplo de demonstração do algoritmo da PAL Eᴛ em Matlab, e sua saída tanto na forma de resposta a uma proposição P, como em um de 12 estados lógicos paraconsistentes pode ser observado em [6].

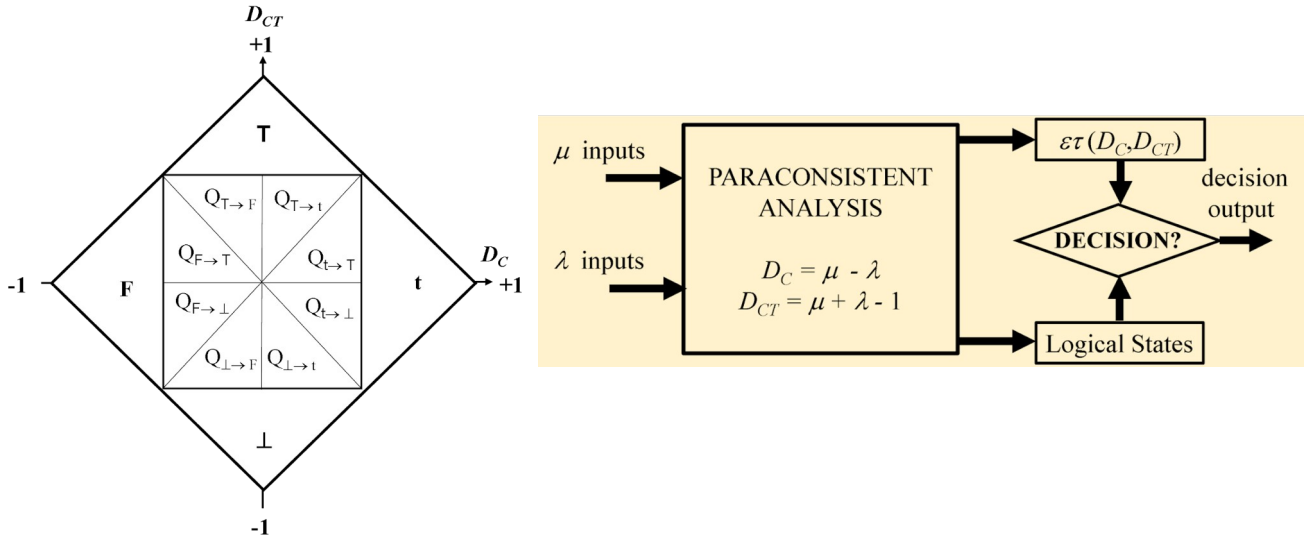


Figura 3 – Divisão do Lattice PAL Eᴛ em 12 estados lógicos e o diagrama de blocos do algoritmo Para-Analyzer, concorrente direto da lógica Fuzzy [3].

3.2. PAL Eᴛ ou Kalman?

Em muitas situações os sensores utilizados em IoT ou em sistemas embarcados são contaminadas por ruídos e imprecisões. O filtro Kalman é um algoritmo bastante utilizado e que produz estimativas dos valores reais de grandezas medidas e valores associados predizendo um valor, estimando a incerteza do valor predito e calculando uma média ponderada entre o valor predito e o valor medido. Entre as vantagens do Filtro Kalman está o fato de ser baseado em sistemas dinâmicos lineares discretizados no domínio do tempo e operar com matrizes, permitindo a construção de sistemas de múltiplas entradas e múltiplas saídas (*multiple input multiple output* – MIMO). Entretanto, há situações em que um simples filtro-passa-

baixas de primeira ou segunda ordem é suficiente para estimar o valor de uma variável. Para esses casos a PAL Eт pode ser uma solução [7,8]. Conforme a Figura 4, a saída atual ($\mu_E[n]$) do PAL2v Filter (outro nome para PAL Eт) consiste basicamente de uma parcela (1-FL) da saída anterior ($\mu_E[n\text{-}1]$) mais uma parcela (F_L) da entrada atual ($\mu[n]$), conforme a equação apresentada na Figura 4, onde F_L corresponder ao fator de aprendizagem e é o ajuste do filtro. Pode-se cascatear várias unidades do PAL2v Filter para criar filtros de ordem superior, como demonstrado na referência [9] disponibilizada em Matlab.

$$\mu E[n] = (1 - FL) * \mu E[n-1] + FL * \mu[n]$$

Figura 4 – Equação básica do Filtro PAL2v de Primeira ordem [7].

3.3. Redes Neurais Artificiais com PAL Eт?

O algoritmo básico da PAL Eт pode ser utilizado para compor a função de ativação de um neurônio. A interligação de vários neurônios PAL Eт podem ser usadas para formar uma rede neural paraconsistente (*paraconsistent neural network* – PNN). Foi demonstrado em [10] que uma PNN apresentou vantagens em relação a redes neurais artificiais (*artificial neural network* – ANN) com funções de ativação clássicas (sigmoide, tanh e ReLU) para a identificação de um sistema dinâmico não linear de pêndulo invertido [10]. Da mesma forma, um sistema de controle por modelo de referência (*model reference control* – MRC), composto por 3 redes neurais paraconsistentes recorrentes (*recurrent paraconsistent neural network* – RPNN) foi implementado com sucesso no controle de um pêndulo invertido rotativo (*rotary inverted pendulum* – RIP), com menor esforço de energia e maior robustez que o sistema clássico de controle por alocação de polos [11]. Esse sistema utilizou apenas 14 neurônios PAL Eт sendo executado em um microcontrolador ESP32 no controle em tempo real do RIP, demonstrando o potencial de utilização de redes neurais paraconsistentes em sistemas embarcados e de *edge computing*.

A Figura 5 apresenta um exemplo de PNN. Uma comparação entre PNN e ANN na forma de algoritmo de Matlab está disponível na referência [12].

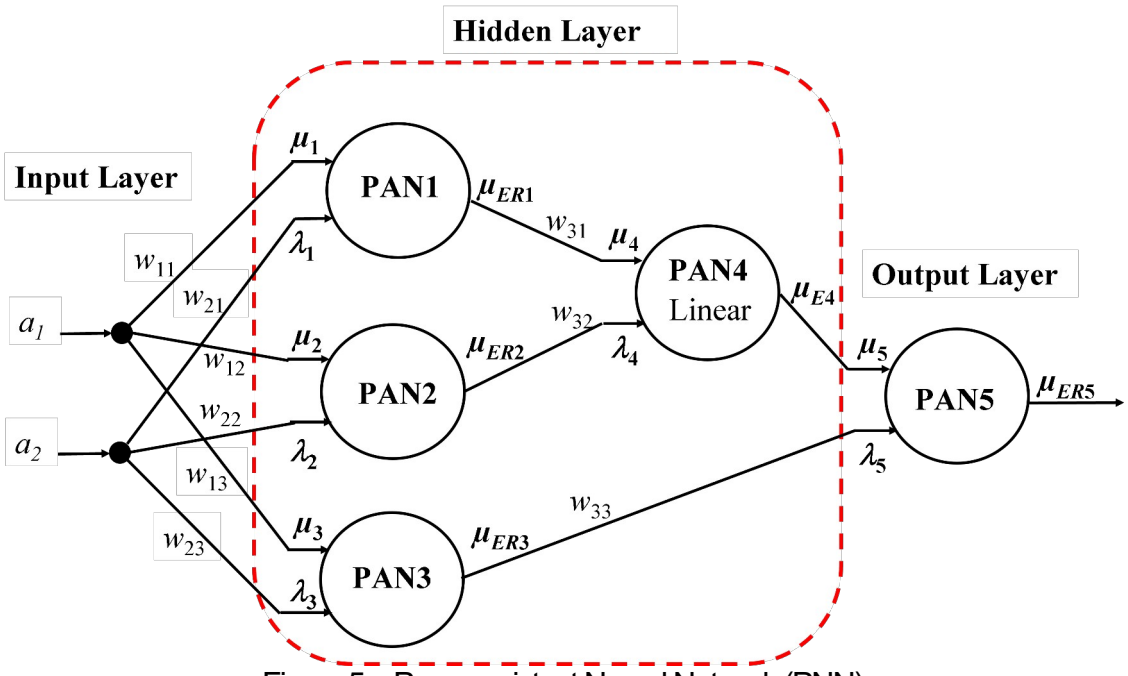


Figura 5 – Paraconsistent Neural Network (PNN)

4. CONSIDERAÇÕES FINAIS

Esse post apresentou alguns algoritmos e suas aplicações bem-sucedidas da PAL Eт na tomada de decisões, filtragem e tratamento de sinal, classificação, reconhecimento de padrões, identificação e controle. A simplicidade das equações e regras da PAL Eт são características importantes e que devem ser levadas em consideração em projetos de sistemas AIoT embarcados e de computação de borda.

Para saber mais sobre a PAL Eт, um bom começo pode ser a pagina “PAL2v: Key Points”, que complementa as referências listadas neste post e está disponível em [13]. Nela há referência aos principais algoritmos, os links para exemplos em Matlab e um conjunto de artigos de referência de alto fator de impacto sobre a lógica.

REFERÊNCIAS

[1] De Carvalho Junior, A. 10 Tendências de Aplicação de Inteligência Artificial em 2024! EAILAB, publicado em nov 28, 2023. Disponível em: https://eailab.labmax.org/2023/11/28/10-tendencias-de-aplicacao-de-inteligencia-artificial-em-2024/

[2] De Carvalho Junior, A. Tendências em Edge Computing. EAILAB, publicado em dez 12, 2023. Disponível em: https://eailab.labmax.org/2023/12/12/tendencias-em-edge-computing/

[3] De Carvalho Junior, A.; Justo, J. F.; De Oliveira, A. M.; Da Silva Filho, J. I., A comprehensive review on paraconsistent annotated evidential logic: Algorithms, Applications, and Perspectives, Engineering Applications of Artificial Intelligence, Volume 127, Part B, 2024, 107342, ISSN 0952-1976 DOI: 10.1016/j.engappai.2023.107342.

[4] Da Silva Filho, J. I. Treatment of Uncertainties with Algorithms of the Paraconsistent Annotated Logic, Journal of Intelligent Learning Systems and Applications, Vol. 4 No. 2, 2012, pp. 144-153. DOI: 10.4236/jilsa.2012.42014. Disponível em: https://www.scirp.org/journal/paperinformation.aspx?paperid=19263. Acessado em dez 21, 2023.

[5] Abe, J.M., Torres, C.R., Lambert-Torres, G., Nakamatsu, K., Kondo, M., 2006b. Intelligent paraconsistent logic controller and autonomous mobile robot emmy II. In: Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 851–857. http://dx.doi.org/10.1007/11893004_108.

[6] De Carvalho Junior, A. (2023). PAL2v Para-Analyzer, MATLAB Central File Exchange. Retrieved November 22, 2023. Disponível em: https://www.mathworks.com/matlabcentral/fileexchange/155422-pal2v-para-analyzer. Acessado em dez 21, 2023.

[7] De Carvalho Jr., A. et al. (2023). A Paraconsistent Artificial Neural Cell of Learning by Contradiction Extraction (PANCLCTX) with Application Examples. In: Abe, J.M. (eds) Advances in Applied Logics. Intelligent Systems Reference Library, vol 243. Springer, Cham. DOI: 10.1007/978-3-031-35759-6_5.

[8] Carvalho, A., Justo, J.F., Angélico, B.A. et al. Paraconsistent State Estimator for a Furuta Pendulum Control. SN COMPUT. SCI. 4, 29 (2023). https://doi.org/10.1007/s42979-022-01427-z

[9] De Carvalho Junior, A. (2023). PAL2v Filter, MATLAB Central File Exchange. Retrieved May 24, 2023. Disponível em: https://www.mathworks.com/matlabcentral/fileexchange/129644-pal2v-filter. Acessado em dez 21, 2023.

[10] Carvalho, A., Justo, J. F., Angélico, B. A. et al, “Rotary Inverted Pendulum Identification for Control by Paraconsistent Neural Network,” in IEEE Access, doi: 10.1109/ACCESS.2021.3080176.

[11] Carvalho, A., Justo, J.F., Angélico, B.A. et al. Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum, Applied Soft Computing, 2022, 109927, ISSN 1568-4946, DOI: 10.1016/j.asoc.2022.109927.

[12] De Carvalho Junior, A. Paraconsistent Neural Network (PNN), MathWorks, Retrieved June 13, 2023. Disponível em: https://www.mathworks.com/matlabcentral/fileexchange/130739-paraconsistent-neural-network-pnn. Acessado em Dez 21, 2023.

[13] De Carvalho Junior, A. PAL2v: Key Points, Google Sites, 2024. Disponível em https://sites.google.com/view/prof-arnaldo/pal2v-key-points. Acessado em Jan 09, 2024.

Elaborado Por: Dr. Arnaldo de Carvalho Junior
Publicado em: Jan 09, 2024
Disponível em: https://eailab.labmax.org/2024/01/09/pal-e%cf%84-a-solucao-para-os-desafios-de-aiot-em-edge-computing/. Acessado em Nov 08, 2024.

Post 4: Função de Ativação, o Núcleo da Composição de Neurônios Artificiais

1. INTRODUÇÃO

As redes neurais artificiais – RNAs (*artificial neural networks* – ANN) são algoritmos poderosos, muito utilizados em aplicações de inteligência artificial (IA) e aprendizado de máquina (*machine learning* – ML). As RNAs são capazes de "aprender" uma determinada função ou reconhecimento de padrões, despertando interesse em diversas áreas do conhecimento humano, de medicina diagnóstica a robótica, automação e controle de sistemas complexos [1].

As RNAs são constituídas de neurônios artificiais cujas funções matemáticas são inspiradas em neurônios biológicos, constituindo a base de redes neurais artificiais. A Figura 1 apresenta um exemplo de interligação de neurônios para compor uma RNA.

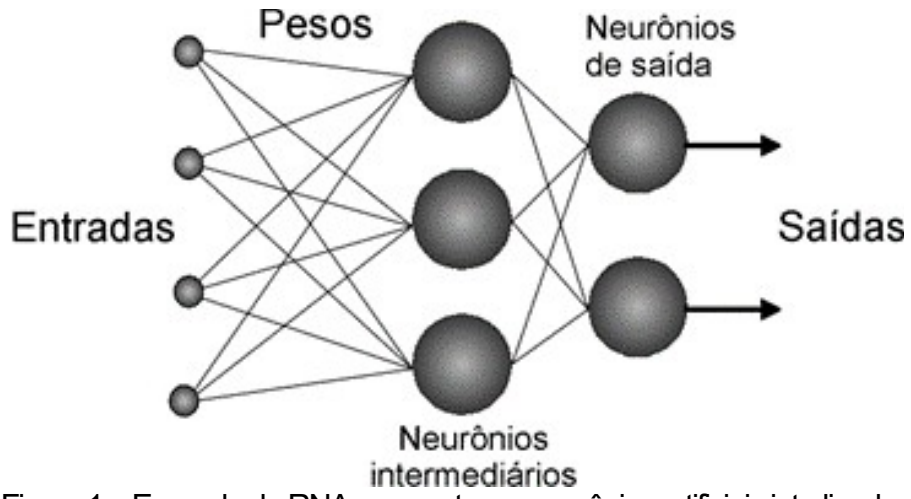


Figura 1 – Exemplo de RNA composta por neurônios artificiais interligados
Fonte: Adaptado de [2]

Apesar de não ter a mesma complexidade de um cérebro, as RNAs apresentam duas similaridades básicas com as redes neurais biológicas [3]:

19. possibilidade de descrição de seus blocos de construção por dispositivos computacionais simples;
20. as conexões entre os neurônios determinam a função da rede.

A Figura 2 apresenta uma representação de um neurônio biológico. Já a Figura 3 apresenta um neurônio artificial de múltiplas entradas.

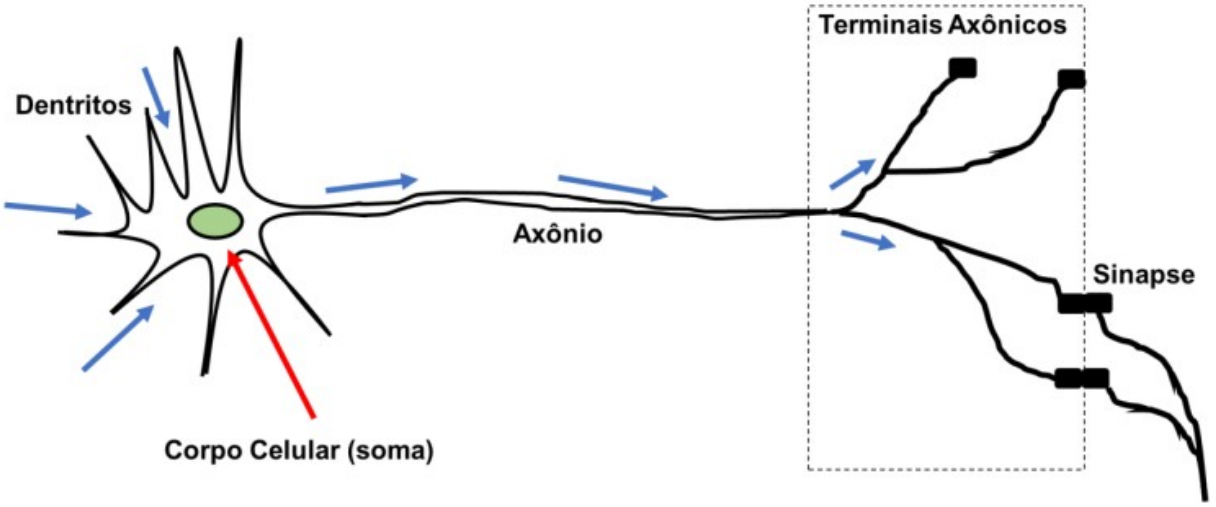


Figura 2 – Representação de um neurônio biológico.
Fonte: Adaptado de [1]

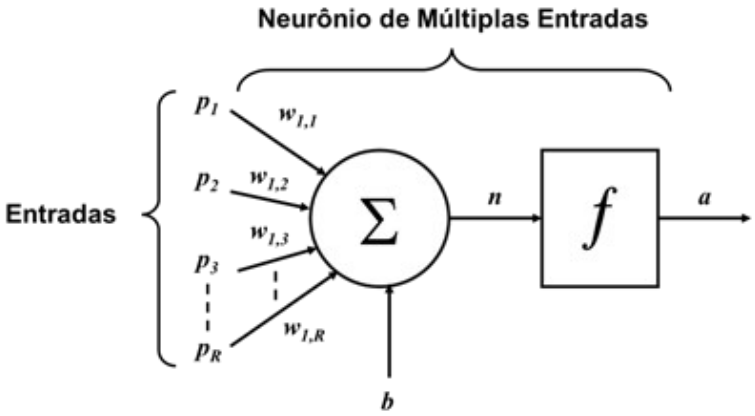


Figura 3 – Representação de neurônio artificial de múltiplas entradas.
Fonte: Adaptado de [1]

No neurônio biológico, os dentritos são filamentos responsáveis por receber os sinais de informações para as células. O corpo celular, ou soma, reúne as informações recebidas pelos dentritos. Já o axônio transporta os impulsos elétricos que partem do corpo celular até as diversas ramificações com dilatações bulbosas conhecidas como terminais axônicos (ou terminais nervosos), que estabelecem as conexões sinápticas com outras células [1].

Conforme a Figura 3, o neurônio artificial consiste de 2 etapas. A primeira é a somatória ponderada dos sinais de entrada, que em seguida são aplicados a uma função de ativação. Os pesos (w) representam a força das sinapses e são multiplicados aos valores das respectivas entradas e somados, juntamente com

um valor de ajuste ou bias (*b*). O resultado desta soma (*n*) é então aplicado a uma função de ativação (*f*) e apresentado na saída (a) do neurônio artificial. Um neurônio artificial pode ser descrito como:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + \cdots + w_{1,R}p_R$$

$$a = f(Wp + b)$$

As funções de ativação são um elemento fundamental das RNAs. Elas essencialmente decidem se um neurônio deve ser ativado ou não. Em outras palavras, se o que o neurônio está recebendo é relevante para a informação fornecida ou deve ser desprezada [4].

Várias funções f podem ser utilizadas, mas é importante que a função de ativação adotada seja derivável, de modo a permitir a elaboração de algoritmos de regressão para calibração, ou "aprendizado", dos pesos e bias do neurônio [3]. Pode ser demonstrado que se a função de ativação for não linear, uma RNA de duas camadas pode ser um aproximador universal de função [5].

2. FUNÇÕES DE ATIVAÇÃO

As Figuras 4 e 5 apresentam algumas das funções de ativação mais utilizadas em projetos de RNAs [6].

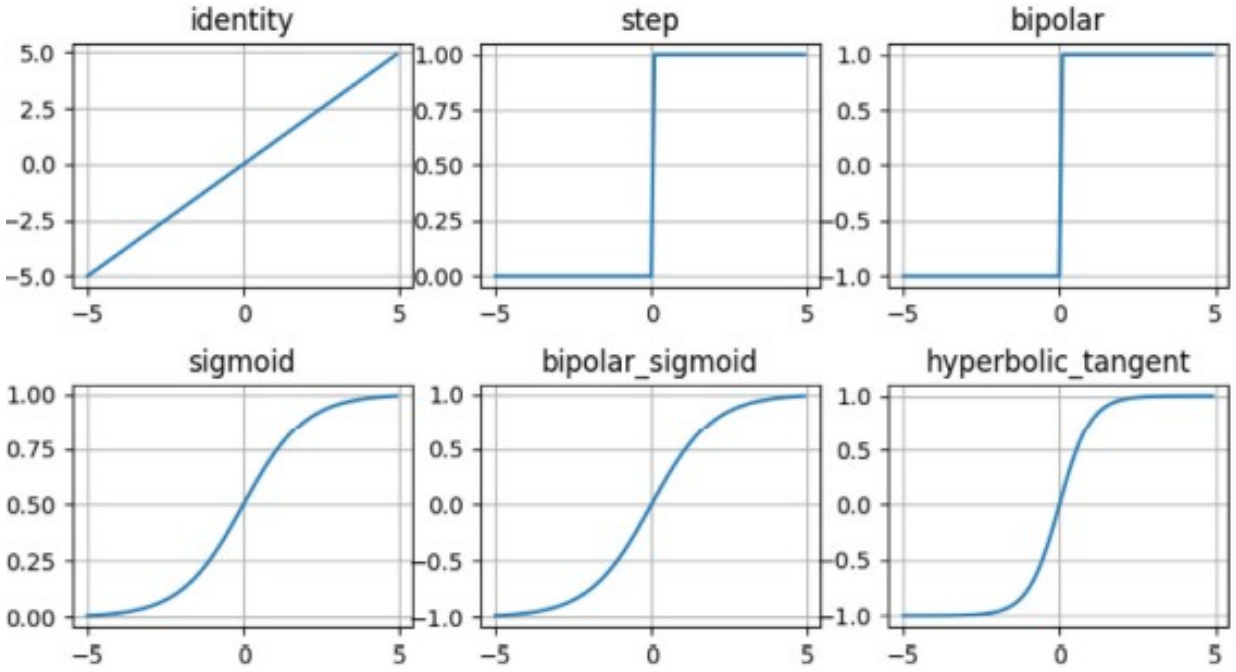


Figura 4 – Funções de Ativação Clássicas
Fonte: Adaptado de [6]

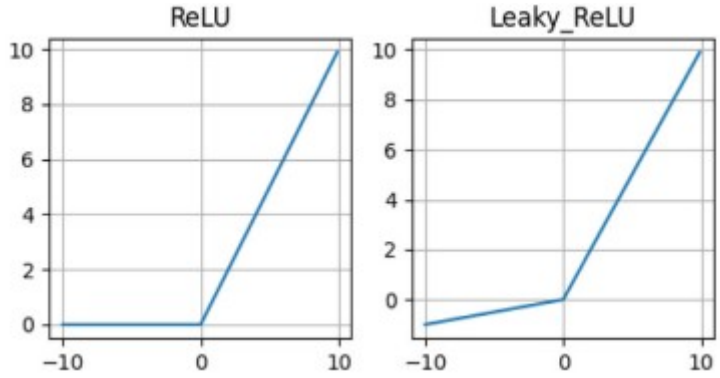


Figura 5 – Funções de Ativação baseadas em Retificadores
Fonte: Adaptado de [6]

2.1. Função Step

A função de ativação mais elementar é a Função Degrau (Step), onde o classificador é baseado em um limiar de ativação (threshold). Ela foi utilizada nos primeiros neurônios artificiais foi introduzido em 1943 por W. McCulloch e W. Pitts [3].

$$f(n) = 1, n \geq 0$$
$$f(n) = 0, n < 0$$
$$f'(n) = 0$$

A função Step é mais teórica do que prática, pois em geral há mais de uma classe de dados para serem classificados. Além disso, o gradiente da função Step é zero, dificultando processos de aprendizagem da RNA.

2.2. Função Linear ou Identidade

A função identidade também é uma função simples, cuja derivada resulta em uma constante (α), não importa o valor da entrada x. Isso dificulta os processos de aprendizagem da RNA. Todavia a função linear pode ser ideal para tarefas simples, onde a interpretabilidade é altamente desejada [4], como por exemplo o neurônio da camada de saída da RNA.

$$f(n) = \alpha n$$
$$f'(n) = \alpha$$

2.3. Função Sigmoide

Uma das funções de ativação mais utilizadas nas RNAs é a log-sigmoide, por ser não linear e pela facilidade de implementação de sua derivada no processo ajuste dos pesos, conforme as equações a seguir [1],[6].

$$f(n) = \sigma(n) = \frac{1}{1 + e^{-n}}$$
$$f'(n) = \sigma(n)(1 - \sigma'(n))$$

A função varia de 0 a 1 tendo um formato S. A função sigmoide essencialmente tenta empurrar os valores de Y para os extremos. Esta é uma qualidade muito desejável quando se deseja classificar os valores para uma classe específica. A função sigmoide é amplamente utilizada. Entretanto, quando os gradientes se tornam muito pequenos, a função se aproxima de zero, dificultando o aprendizado do neurônio. Sua derivada tende a 0 para valores de entrada maiores que +5 e abaixo de -5. Pelos valores de saída estarem limitados a (0,1), pode exigir algum tipo de normalização dos sinais de entrada para que sejam sempre positivos. Além disso, em geral, uma RNA composta por sigmoide utiliza mais ciclos de aprendizado do que RNAs com funções de ativação mais "rápidas" [1],[4].

2.4. Função Tangente Hiperbólica (*tanh*)

A *tanh* é outra função do tipo sigmoide. Na verdade, é apenas uma versão escalonada da função sigmóide., variando entre -1 e +1, cuja expressão e derivada são dadas pelas equações a seguir:

$$f(n) = a = \tanh(n) = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$$
$$f'(n) = \tanh'(n) = 1 - \tanh^{2}(n)$$

A *tanh* funciona de forma semelhante à função sigmóide, mas é simétrica em relação à origem, variando entre (-1,1). Ela Basicamente, soluciona o problema dos valores, sendo todos do mesmo sinal. Todas as outras propriedades são as mesmas da função sigmoide. É contínua e diferenciável em todos os pontos. A função não é linear, resultando em algoritmos de treinamento ligeiramente mais rápidos que a log-sigmoide [1],[4].

2.5. Funções Baseadas em Retificadores

A função ReLU é a unidade linear retificada (retified linear unit – ReLU). A função ReLU é muito parecida com a função identidade, fazendo com o processo de aprendizagem da RNA, baseado nessa função de ativação, seja muito mais rápido que por sigmoides [7].

$$a = \mathrm{ReLU}(n) = \begin{cases} 0 & n \leq 0 \\ n & n > 0 \end{cases}$$

$$f'(n) = \mathrm{ReLU}'(n) = \begin{cases} 0 & n \leq 0 \\ 1 & n > 0 \end{cases}$$

A ReLU é uma das funções de ativação mais utilizadas atualmente. A principal vantagem de usar a função ReLU é que ela não ativa todos os neurônios ao mesmo tempo. Na função ReLU, se a entrada for negativa, ela será convertida em zero e o neurônio não será ativado. Isso significa que, ao mesmo tempo, apenas alguns neurônios são ativados, tornando a rede esparsa, eficiente e de computação fácil. Essa vantagem também pode ser considerada uma desvantagem, pois os neurônios utilizando ReLU tendem a "morrer" durante o treinamento, causando a saída do neurônio iniciar a produzir apenas zeros. Uma variação da ReLU, chamada Leaky-ReLU (LReLU) evita isso [7] cuja função e sua derivada são apresentadas pelas equações a seguir:

$$a = \mathrm{LReLU}(n) = \begin{cases} \alpha n & n \leq 0 \\ n & n > 0 \end{cases}$$

$$\mathrm{LReLU}'(n) = \begin{cases} \alpha & n \leq 0 \\ 1 & n > 0 \end{cases}$$

Onde α é um parâmetro introduzido na LReLU, com valores propostos entre 0.01 e 0.2 [7]. A LReLU permite o neurônio ter um pequeno gradiente quando ele não está ativo ($n < 0$), reduzindo o problema potencial mencionado sobre ReLU [6].

2.6. Neurônio PAL2v

O neurônio PAL2v utiliza o algoritmo da lógica paraconsistente anotada com anotação de 2 valores (*paraconsistent annotated logic by 2-value annotations* – PAL2v), também chamada de lógica paraconsistente anotada evidencial (paraconsistent annotated evidential logic – PAL $\varepsilon\tau$) [1]. A PAL2v é uma variação da lógica paraconsistente, proposta pelo matemático brasileiro Newton da Costa [1].

O neurônio PAL2v, quando entradas ponderadas por peso são combinadas e aplicadas à função de ativação PAL2v foi proposto em [1] e aplicado com sucesso na identificação e controle de sistemas dinâmicos não lineares [8],[9]. Ao contrário das funções de ativação que possuem uma entrada (n) e uma saída (a), o neurônio PAL2v possui 2 entradas ortogonais entre si (μ,λ). Esta característica dá uma flexibilidade muito grande ao neurônio PAL2v. Os sinais de interesse podem ser aplicados apenas a uma entrada enquanto que a outra funciona como um "*bias*" na função de ativação. Ou uma das entradas pode receber os sinais de interesse com pesos enquanto que a outra entrada funciona como realimentação da saída, para análise de séries temporais, como em redes neurais recorrentes (*recurrent neural networks* – RNN) [1],[9]. A Figura 6 apresenta um diagrama conceitual da função de ativação PAL2v.

Por ser baseada em uma lógica, os valores de μ e λ são limitados entre (0,1), por isso na figura aparece uma saturação entre esses dois valores, antes da operação da função PAL2v.

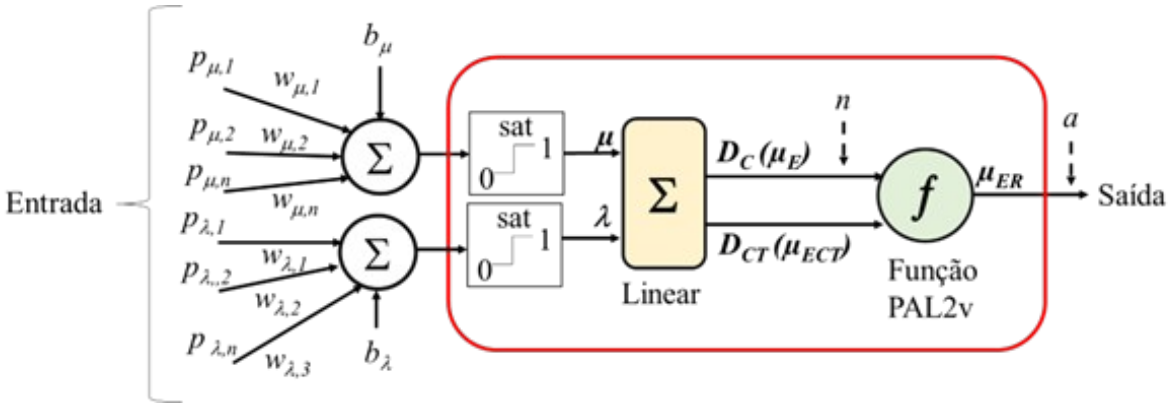


Figura 6 – Função de Ativação Neurônio PAL2v
Fonte: Adaptado de [1].

A função PAL2v de forma simplificada pode ser calculada conforme a seguir:

$$D = \sqrt{(1-|\mu-\lambda|)^2 + (\mu+\lambda-1)^2}$$

$$D = \begin{cases} 1 & D \geq 1 \\ D & 0 \leq D < 1 \end{cases}$$

$$\mu_{ER} = \begin{cases} \dfrac{2-D}{2} & (\mu-\lambda) > 0 \\ \dfrac{D}{2} & (\mu-\lambda) < 0 \\ 0 & (\mu-\lambda) = 0 \end{cases}$$

Interessante que a função de ativação PAL2v apresenta um comportamento dual. Se uma das entradas for mantida em 0.5, a saída será uma sigmoide. Do contrário a saída poderá saturar em 0.5, apresentando uma curva tipo retificadora não linear. A figura 7 apresenta a saída da função de ativação PAL2v, variando-se μ e mantendo-se λ constante Já a Figura 8 apresenta a saída da função de ativação PAL2v mantendo-se μ constante e variando-se λ [8],[9].

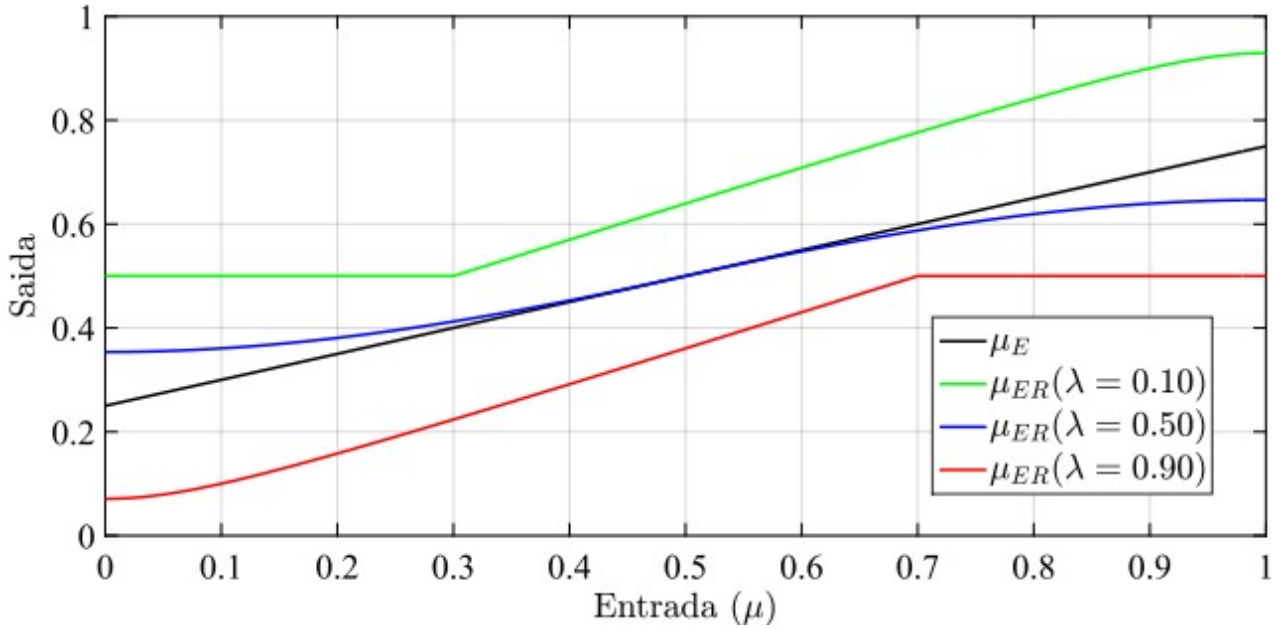


Figura 7 – Saída da Função PAL2v, variando-se μ.
Fonte: Adaptado de [1].

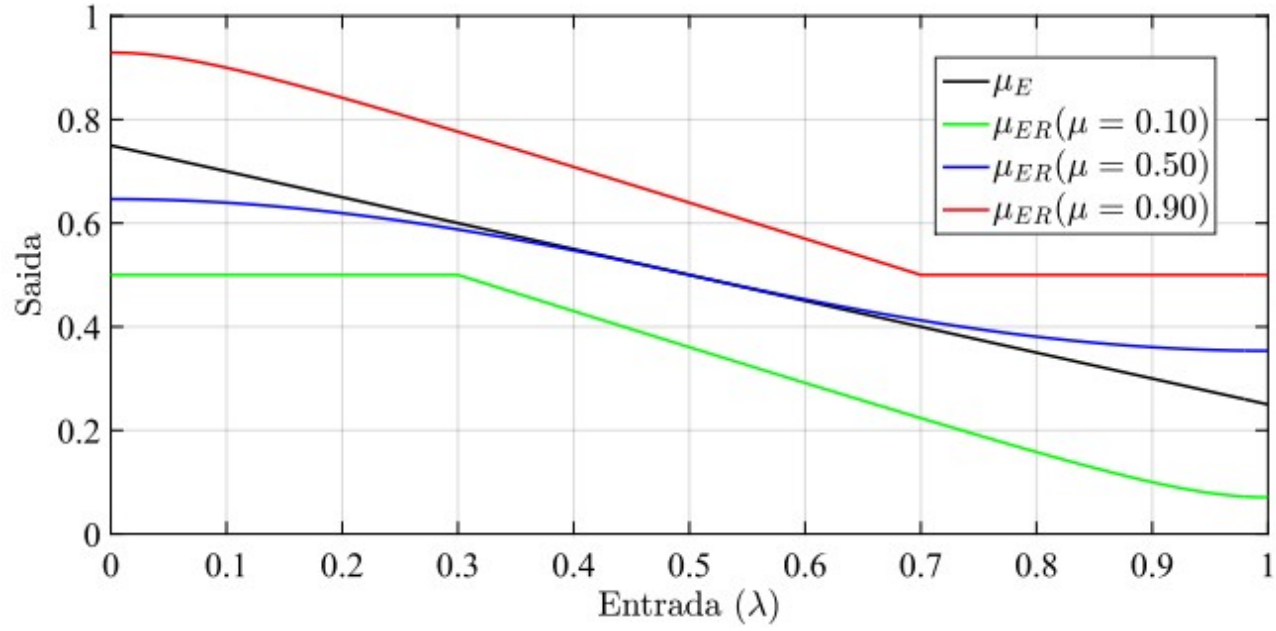


Figura 8 – Saída da Função PAL2v, variando-se λ.
Fonte: Adaptado de [1].

O neurônio PAL2v apresentou melhor erro médio quadrático (mean square error – MSE) e menos ciclos de treinamento que RNAs equivalentes com funções sigmoide, tangente hiperbólica e LReLU para aplicações de identificação e controle de pêndulo invertido rotativo [8],[9], tanto com RNA convencional do tipo feed-forward, como em RNN [8],[9].

Uma comparação entre redes neurais utilizando as funções de ativação sigmoide, tanh, ReLU, LReLU e PAL2v em Matlab está disponível em [10].

2.7. Outras Funções de Ativação

Como dito anteriormente, existem muitas outras funções de ativação. Uma classe especial são as funções base radial (*radial basis functions* – RBF). Nessa categoria há uma grande quantidade de funções, tais como:

a. Gaussiana

$$\varphi(r) = e^{-(\varepsilon r)^2},$$

b. Multiquadrática

$$\varphi(r) = \sqrt{1 + (\varepsilon r)^2},$$

c. Inversa da Multiquadrática

$$\varphi(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}},$$

As RBFs são aproximadores universais muito extremamente eficientes. O treinamento de rede neural RBF (RBF *neural network* – RBFNN) é mais rápido que funções de ativação do tipo sigmoide. A desvantagem da RBFNN está na sua complexidade, que aumenta conforme o crescimento de neurônios na camada oculta. Outro desafio da RBF está em sua estrutura algoritmo de treinamento, não permitindo modela um sistema fortemente não linear [9]. Na literatura há uma grande variedade de RBFs propostas [11].

3. COMENTÁRIOS FINAIS

Este artigo procurou apresentar de forma suscinta os tipos de funções de ativação mais comuns utilizados em projetos de RNAs e suas características básicas, além de propostas que tem recebido atenção dos pesquisadores, como a função PAL2v e as RBFs.

A escolha da função de ativação da RNA passa por diversas questões tais como a complexidade da função, os algoritmos de aprendizagem da RNA, se o sistema resultará em “neurônios mortos”, se o processo de aprendizagem é suave ou apresenta dissipação do gradiente (*Vanishing Gradient*), qual o poder computacional exigido para o treinamento da RNA, quantos neurônios e quantas camadas são necessários, entre outros fatores.

4. REFERÊNCIAS

[1] DE CARVALHO JUNIOR, A. Identificação e Controle de Sistemas Dinâmicos com Rede Neural Paraconsistente. 2021. 196 p. Tese (Doutorado) – Programa de Engenharia Elétrica, Escola Politécnica, Universidade de São Paulo, São Paulo, 2021. Disponível em https://www.teses.usp.br/teses/disponiveis/3/3142/tde-08102021-100149/pt-br.php. Acessado em fevereiro 27, 2024.

[2] TAFNER, M. A. O que são as Redes Neurais Artificiais, Revista Cerebro e Mente 2(5), 1998. Disponível em https://cerebromente.org.br/n05/tecnologia/rna_i.htm, acessado em fevereiro 27, 2024.

[3] HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. H.. Neural Network Design, Martin Hagan; 2º edition, 2014, 802 p. Disponível em https://hagan.okstate.edu/NNDesign.pdf, acessado em fevereiro 27, 2024.

[4] DSA, Função de Ativação, Deep Learning Book, Data Science Academy. Disponível em https://www.deeplearningbook.com.br/funcao-de-ativacao, acessado em fevereiro 27, 2024.

[5] SONODA, S.; MURATA, N. Neural network with unbounded activation functions is universal approximator. Applied and Computational Harmonic Analysis, Vol. 43, Issue 2, 2017, p. 233-268. DOI: 10.1016/j.acha.2015.12.005.

[6] APICELLA, A.; DONNARUMMA, F.; ISGRÒ, F.; Prevete, R. A survey on modern trainable activation functions, Neural Networks, Volume 138, p. 14-32, 2021. DOI: 10.1016/j.neunet.2021.01.026.

[7] LIU, X.; JIA, R.; LIU, Q.; ZHAO, C. AND SUN, H. Coastline Extraction Method Based on Convolutional Neural Networks—A Case Study of Jiaozhou Bay in Qingdao, China, in IEEE Access, vol. 7, p. 180281-180291, 2019. DOI: 10.1109/ACCESS.2019.2959662.

[8] A. De Carvalho, J. F. Justo, B. A. Angélico, A. M. De Oliveira and J. I. d. S. Filho, “Rotary Inverted Pendulum Identification for Control by Paraconsistent Neural Network,” in IEEE Access, doi: 10.1109/ACCESS.2021.3080176.

[9] Carvalho, A., Justo, J.F., Angélico, B.A. et al. Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum, Applied Soft Computing, 2022, 109927, ISSN 1568-4946, DOI: 10.1016/j.asoc.2022.109927.

[10] CARVALHO, A. Paraconsistent Neural Network (PNN). MATLAB Central File Exchange. Retrieved June 13, 2023. Disponível em https://www.mathworks.com/matlabcentral/fileexchange/130739-paraconsistent-neural-network-pnn, acessado em fevereiro 27, 2024.

[11] DASH, Ch. Sanjeev Kumar; et al. Radial basis function neural networks: a topical state-of-the-art survey. Open Computer Science, vol. 6, no. 1, 2016, pp. 33-63. DOI: 10.1515/comp-2016-0005.

Elaborado Por: Dr. Arnaldo de Carvalho Junior

Publicado em: Fev 28, 2024

Disponível em: https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-artificiais/. Acessado em: Nov 08, 2024.

Post 5: Redes Neurais Artificiais: Algoritmos poderosos para aplicações de IA e ML

1. INTRODUÇÃO

A inteligência artificial – IA (*artificial intelligence* – AI), segundo a Encyclopædia Britannica, é uma disciplina da ciência da computação que pesquisa a capacidade de um recurso computacional executar tarefas comumente associadas a seres inteligentes. O termo é amplo e geralmente aplicado ao desenvolvimento de sistemas e processos com capacidade de raciocinar, classificar, descobrir significado, generalizar ou aprender com experiências passadas [1]. A Inteligência Artificial pode ser dividida em 7 subcampos, conforme Figura 1 [2]:

- Aprendizado de máquina (*machine learning* – ML), Mineração de Dados e Grandes Dados (*Data Mining e Big Data*);
- Planejamento Automatizado;
- Sistemas Especialistas (*expert systems*);
- Processamento de Linguagem Natural (*natural language processing* – NLP);
- Reconhecimento de fala (*speech recognition*);
- Robótica;
- Visão Computacional (*computer vision*).

A ML é um ramo da IA (Figura 2), que se preocupa com a implementação de software de computador capaz de aprender autonomamente. Sistemas especialistas e programas de mineração de dados são as aplicações mais comuns para melhorar algoritmos através do uso de aprendizado de máquina [3].

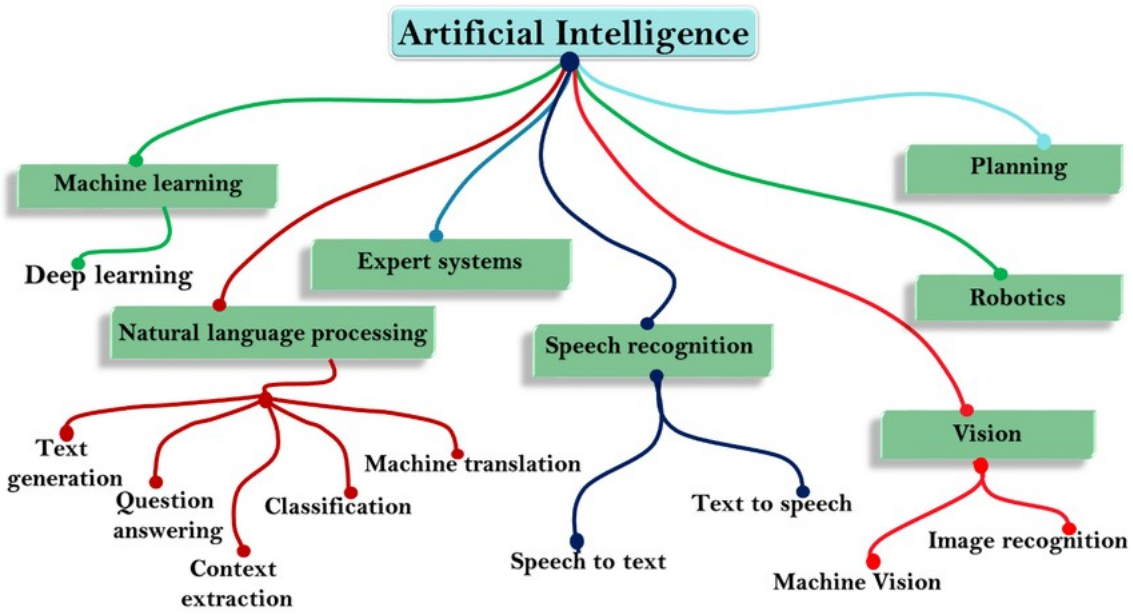


Figura 1 – Subcampos da IA [2]

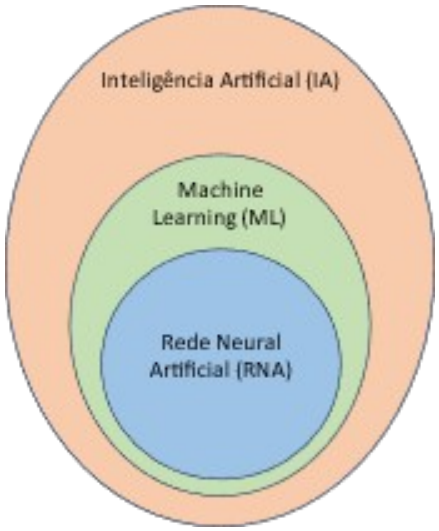


Figura 2 – Relação entre IA, ML e RNA.

As redes neurais artificiais – RNAs (*artificial neural networks* – ANN) são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência [4]. Essa estrutura é formada por neurônios artificiais, funções de ativação específicas, conforme apresentado em [5]. As RNAs desempenham papel fundamental nos algoritmos de ML e IA.

2. TIPOS DE REDES NEURAIS ARTIFICIAIS

O aumento do poder computacional das últimas décadas permitiu uma rápida evolução e proposição de diferentes arquiteturas de RNAs para as mais variadas áreas do conhecimento humano [5]. A Figura 3 fornece uma ideia da variedade de configurações de RNAs [6].

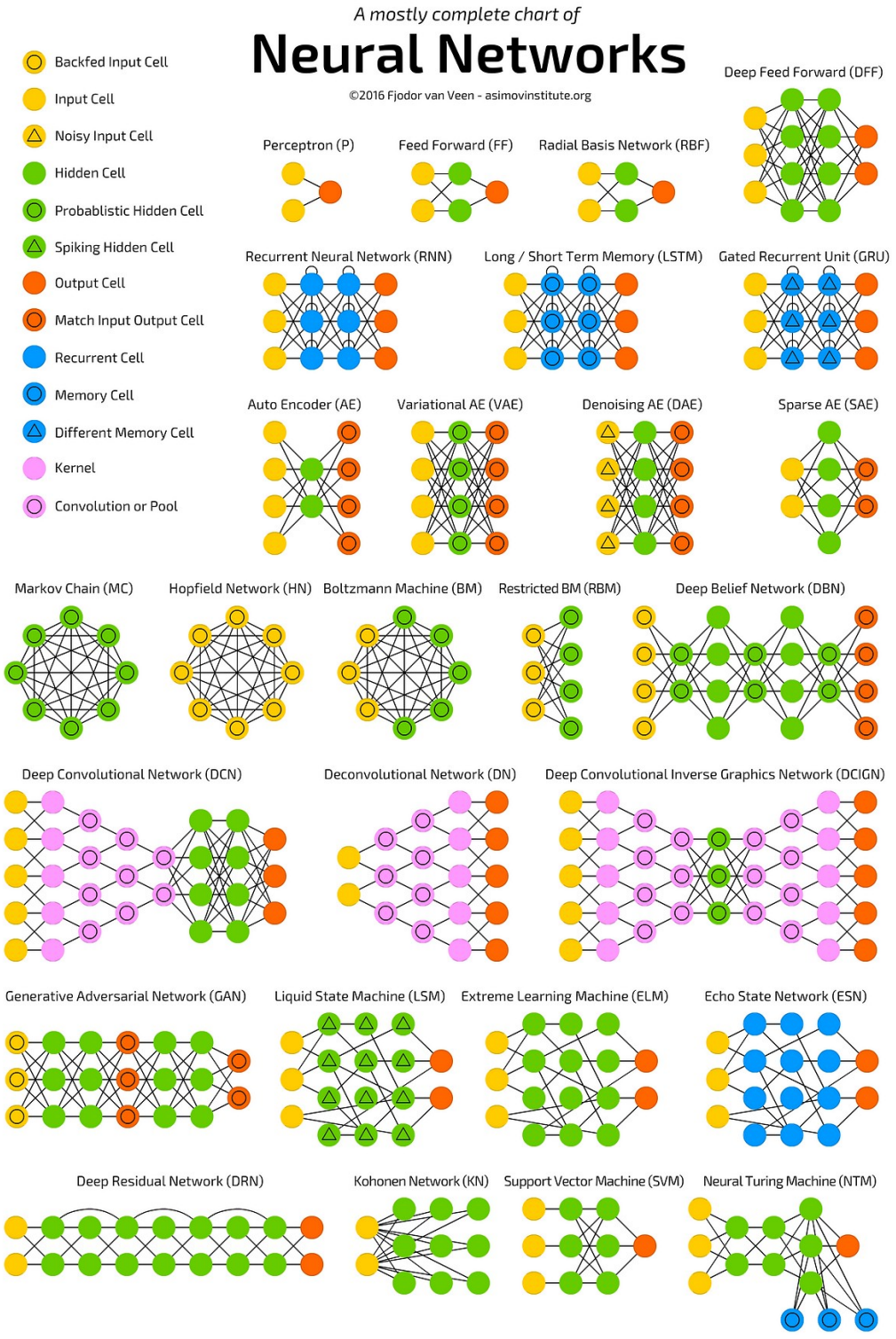


Figura 3 – Tipos de RNAs [6].

Neste post serão apresentados as RNAs mais comumente empregadas em aplicações de ML.

2.1. Rede Neural Feedforward

A RNA do tipo feedforward foi uma das primeiras propostas, sendo uma das mais básicas. Nesta RNA, os dados ou a entrada fornecida viajam em uma única direção (forward). Os dados entram na RNA pela camada de entrada (input layer) e saem pela camada de saída (output layer), enquanto camadas ocultas (hidden layer) podem ou não existir. Logo, a rede neural feedforward (FNN) possui apenas uma onda propagada frontalmente sendo geralmente treinadas utilizando-se o método de retropropagação. Podem ser usadas para aproximação de funções e classificação de padrões [6].

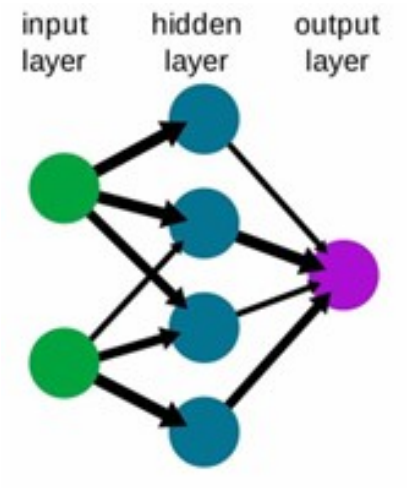


Figura 4 – FNN.

2.2. Rede Neural de Base Radial

A rede neural de função base radial (*radial basis function neural network* – RBFNN) é um tipo de RNA feedforward que faz uso de funções de base radial como funções de ativação nos neurônios da camada oculta. Uma função de base radial é uma função de valor real, cujo valor depende apenas da distância da origem. Embora existam vários tipos de funções de base radial, a função Gaussiana é a mais frequentemente empregada, conforme Eq. 1, a seguir.

$$h_j(x) = \exp\left(-\|x - c_j\|^2 / \sigma_j^2\right) \quad (1)$$

Onde (x) é o vetor de entrada, (c_j) é o centro da função Gaussiana e $h_j(x)$ a saída do neurônio *j*. A saída da rede é uma combinação linear de funções de base radial das entradas e parâmetros do neurônio. As RBFNN têm muitos usos, incluindo aproximação de funções, previsão de séries temporais, classificação e

controle de sistema. O treinamento da RBFNN também é mais rápido do que a FNN utilizando neurônios baseados em sigmoide. Outras características da RBFNN incluem design fácil, boa generalização e robustez para ruído de entrada. Para problemas de aproximação de função, as RBFNN são especialmente recomendadas para superfícies com picos e vales regulares. Uma desvantagem da RBFNN é a sua complexidade, que aumenta à medida que o número de neurônios na camada oculta aumentam. Outro desafio da RBF está no seu algoritmo de treinamento e estrutura, não permitindo modelar um ambiente de sistema fortemente não linear [4].

2.3. Rede Neural Recorrente

A rede neural recorrente (*recurrent neural network* – RNN) difere da FNN pela direção do fluxo de informações entre as suas camadas. A RNN é uma rede neural artificial bidirecional, pois permite que a saída de alguns nós afete a entrada subsequente para os mesmos nós. Em outras palavras, as RNNs são redes com loops, permitindo que as informações persistam por mais tempo na rede. Sua capacidade de usar o estado interno (memória) para processar sequências arbitrárias de entradas torna as RNN aplicáveis a tarefas como reconhecimento de escrita manual conectada e não segmentada ou reconhecimento de fala.

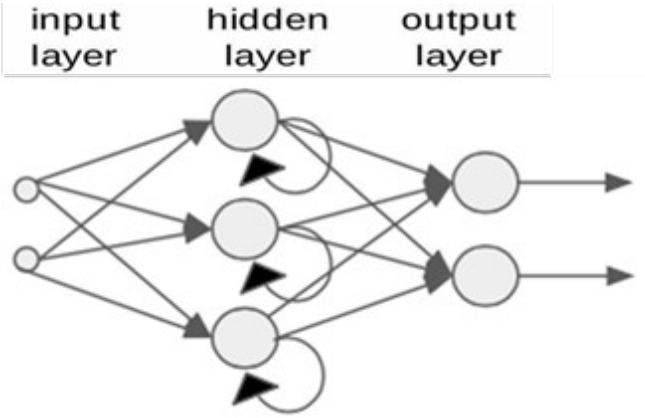


Figura 5 – RNN.

As RNN são muito utilizadas para aplicações que utilizam séries temporais como entrada, como na modelagem e controle de sistemas não lineares, apesar das dificuldades maiores de treinamento do que redes FNN. Um exemplo é o desenvolvimento de um controle por modelo de referência (*model reference control* – MRC), utilizando RNN com neurônios PAL2v (*recurrent paraconsistente neural network* – RPNN) para prever a saída futura do controlador de um sistema dinâmico de pêndulo invertido rotativo, apresentando maior robustez e menor esforço do que um controle convencional, em [4].

2.4. LSTM

A rede neural de memória de longo e curto prazo (long short-term memory – LSTM) é uma variação de rede neural recorrente (RNN), que visa lidar com o problema do gradiente de fuga (*vanishing gradient*) presente em RNNs tradicionais. A unidade LSTM possui uma estrutura em cadeia composta por diferentes blocos de memória chamados células (cells), conforme a Figura 6. As células retêm a informação, enquanto os portões (gates) realizam a manipulação da memória. O cascateamento de unidades LSTM foram a rede neural LSTM, conforme a Figura 7.

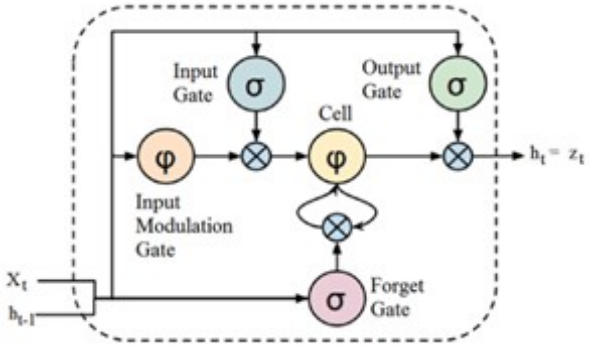


Figura 6 – unidade LSTM [8].

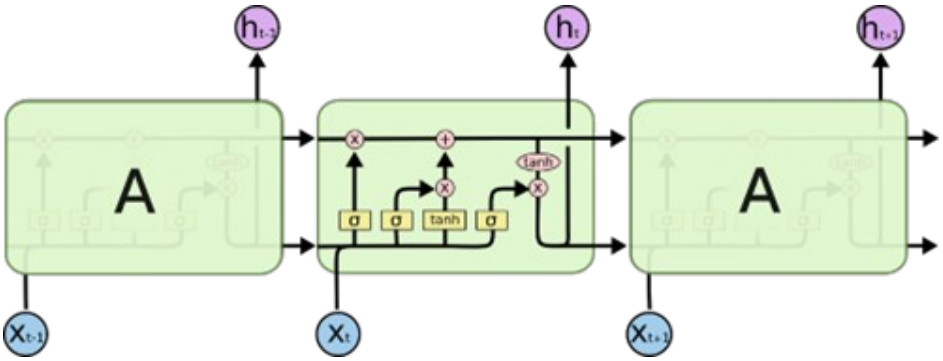


Figura 7 – Rede Neural LSTM [9].

As células retêm a informação, enquanto os portões (*gates*) realizam a manipulação da memória [8]. Os *gates* são:

a) Esqueça o portão (*Forget Gate*): As informações que não são mais úteis são removidas com o forget gate. A entrada no momento específico (x_t) e a saída da célula LSTM anterior (h_{t-1}) são alimentadas ao gate e multiplicadas por matrizes de peso, seguidas da adição de um *bias*, cujo resultado é aplicado a uma função de saída binária [5]. Se para um determinado estado de célula a saída for "0", a informação é esquecida e para a saída "1", a informação é retida para uso futuro.

b) Portão de entrada (*Input Gate*): A adição de informações úteis ao estado da célula é feita pelo *input gate*. Inicialmente, a informação é regulada usando a função sigmoide [5] que filtra os valores a serem lembrados usando as entradas x_t e, de maneira parecida com o *forget gate*. Então, mediante uma função de ativação tanh [5] um vetor é criado, fornecendo uma saída limitada a [-1, +1], que contém todos os valores possíveis de x_t e h_{t-1}. Os valores do vetor e os valores regulados são multiplicados para obter as informações úteis.

c) Portão de saída (*Output Gate*): A tarefa de extrair informações úteis do estado da célula atual para ser apresentadas como uma saída é feita pelo *output gate*. Primeiro, um vetor é gerado aplicando a função de ativação tanh na célula. Então, a informação é regulada usando a função de ativação sigmoide que filtra os valores a serem lembrados usando as entradas x_t e h_{t-1}. Os valores do vetor e os valores regulados são multiplicados para serem enviados como uma saída e entrada para a próxima célula.

A LSTM é adequada para classificar, processar e prever séries temporais com intervalos de tempo de duração desconhecida, devido a sua capacidade de lembrar de valores em intervalos arbitrários. Além de aplicações que envolvem predição de saída futura, as redes LSTM têm sido utilizadas com sucesso na modelagem de linguagem, tradução de idiomas, legendas de imagens, geração de texto e chatbots.

2.5. Rede Neural Convolucional

A rede neural convolucional (*convolucional neural network* – CNN) é um tipo de FNN que aprende uma característica por meio de otimização de filtros. Gradientes de fuga e gradientes de explosão, vistos durante a retropropagação em redes neurais clássicas, são evitados pelo uso de pesos regularizados em menos conexões. Por exemplo, para processar uma imagem de 100×100 pixels, seriam necessários 10.000 pesos para cada neurônio na camada totalmente conectada. No entanto, aplicando kernels de convolução em cascata, apenas 25 neurônios são necessários para processar blocos de tamanho 5×5. Os recursos da camada superior são extraídos de janelas de contexto mais amplas, em comparação com os recursos da camada inferior.

Várias técnicas de convolução podem ser aplicadas, procurando extrair características específicas de um padrão 2D como contorno, contraste, objeto, etc. A Figura 8 apresenta um exemplo de convolução pela média, reduzindo uma matriz 4×4 (16 pixels) em uma matriz 2×2 (4 pixels).

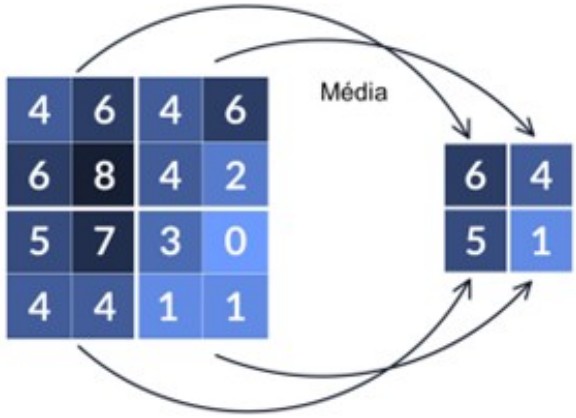


Figura 8 – Convolução pela Média.

A Figura 9 apresenta uma arquitetura de CNN [10]. Pode-se observar que uma CNN é composta por uma ou mais camadas convolucionais (*hidden layers*), seguidas por uma ou mais camadas de neurônios totalmente conectados, para a etapa de classificação, como uma rede FNN padrão.

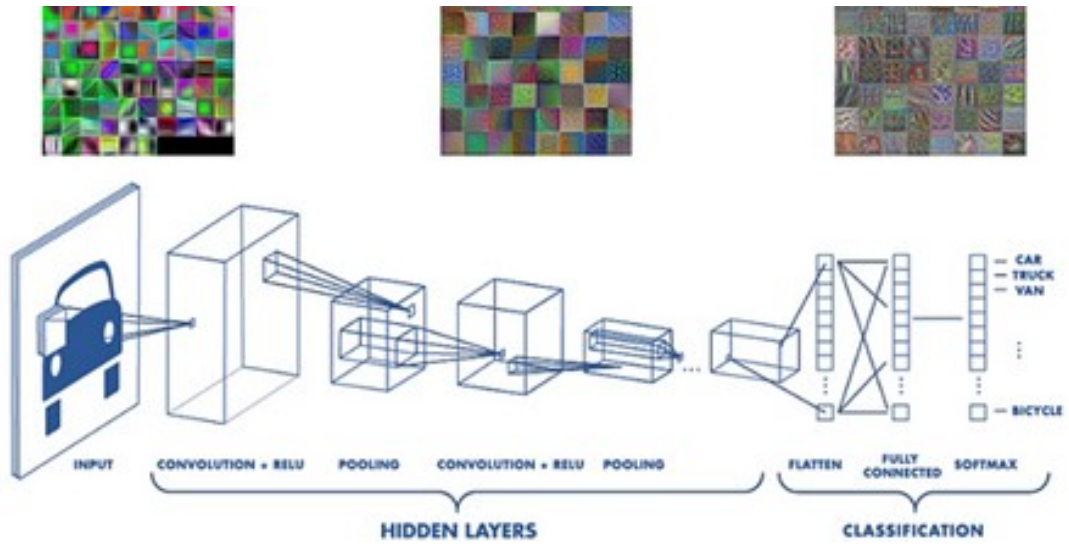


Figura 9 – Arquitetura de CNN [10].

A arquitetura de uma CNN é projetada para aproveitar a estrutura 2D de uma imagem de entrada (ou de um sinal contínuo transformado em padrão 2D sendo invariantes à mudança (*shift invariant*) ou invariantes em espaço (*space invariant*). Outro benefício das CNNs é que elas são mais fáceis de treinar e possuem muito menos parâmetros do que redes totalmente conectadas com o mesmo número de unidades ocultas.

Aplicações de CNN incluem reconhecimento, segmentação e classificação de imagem e vídeo; sistemas de recomendação; análise de imagem médica; processamento de linguagem natural; interfaces cérebro-computador e séries temporais como financeiras, por exemplo. Uma ferramenta que utiliza CNN para

identificação e classificação de imagens é o Edge Impulse [11]. Exemplos de aplicação podem ser observados nos artigos relacionados em [12].

2.6. Redes Neurais de Aprendizado Profundo

Redes Neurais de aprendizado profundo (*deep learning neural network* – DLNN) ou simplesmente redes neurais profundas (*deep neural network* – DNN) são redes neurais (FNN, RNN, CNN, entre outras) que possuem múltiplas camadas intermediárias. Embora possam existir diferentes topologias, a maioria dos modelos modernos de aprendizagem profunda são baseados em CNN. A Figura 10 apresenta uma arquitetura de DNN.

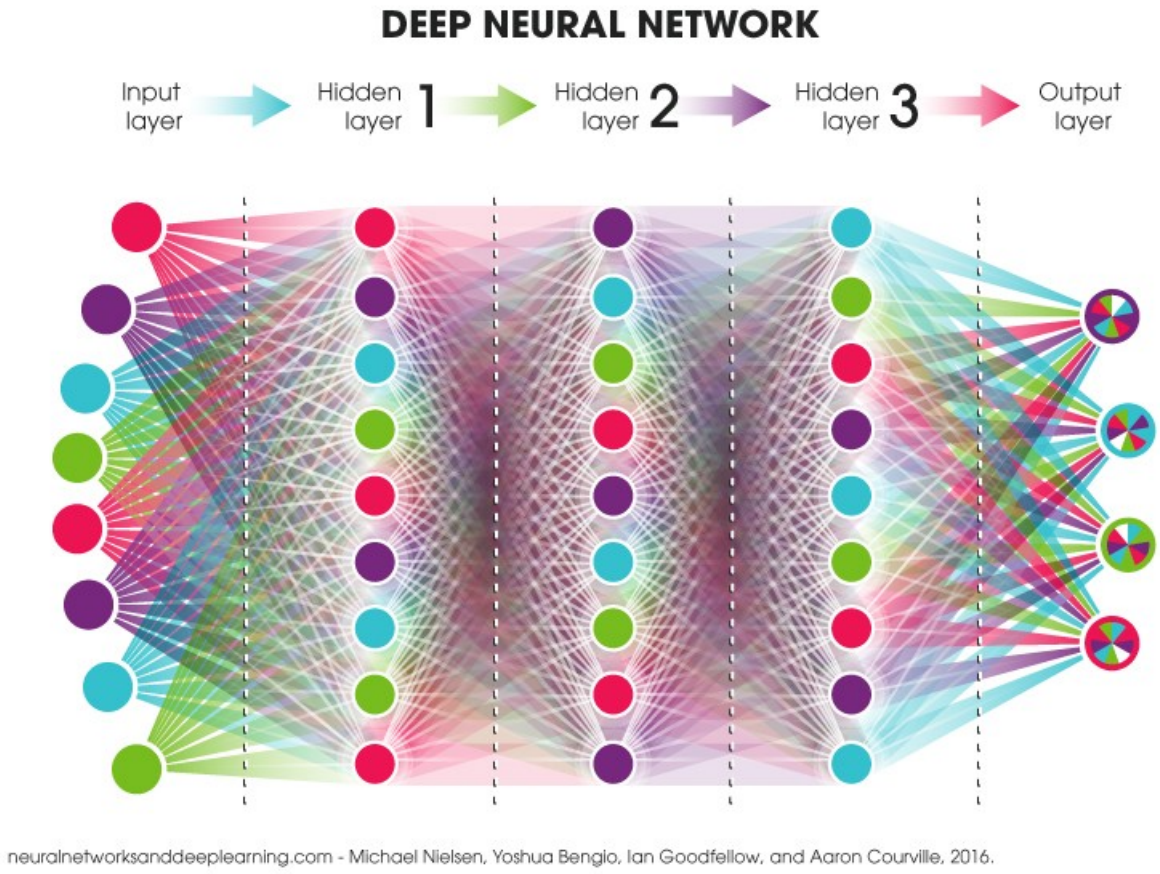


Figura 10 – Arquitetura de DNN [14].

O aprendizado profundo é uma classe de algoritmos de aprendizado de máquina que utiliza múltiplas camadas para extrair progressivamente recursos de nível superior da entrada bruta. Um exemplo de aplicação de DNN é no processamento de imagens, onde as camadas inferiores podem identificar bordas e contornos, enquanto as camadas superiores podem identificar os conceitos relevantes para um ser humano, como dígitos, letras ou rostos.

Como pode ser observado pela Figura 10, as DNNs são mais complexas que as redes neurais convencionais, com algoritmos mais densos e exigindo maior poder de processamento para o seu treinamento. Entretanto, as DNNs têm sido amplamente utilizadas no reconhecimento de imagens, processamento de linguagem natural, predições financeiras, conversão de textos em imagens, entre outras aplicações. Uma boa revisão sobre algoritmos de aprendizado profundo e suas aplicações está disponível em [15].

3. CONSIDERAÇÕES FINAIS

As redes neurais artificiais são um poderoso instrumento para o sucesso do machine learning e da inteligência artificial. Este é um ramo da ciência da computação em acelerada evolução, com uma variedade virtualmente infinita de RNAs propostas. Este post apresentou algumas das arquiteturas de RNAs mais utilizadas, as suas características principais e potencial de aplicações em inteligência artificial. Em resumo, as FNN podem ser usadas em aproximação de funções e classificação de sinais, as RNN e suas variações (LSTM, por exemplo) utilizam séries temporais para previsão de saída futura. Já as CNN e suas variações são utilizadas em visão computacional, como no tratamento e classificação de imagens.

REFERÊNCIAS

[1] COPELAND, B.J. Artificial Intelligence. Encyclopedia Britannica, 1 Apr. 2024. Disponível em: https://www.britannica.com/technology/artificial-intelligence. Acessado em Abr 1, 2024.

[2] KAYID, A. The role of Artificial Intelligence in future technology. Department of Computer Science, The German University in Cairo, 2020. Disponível em: The-role-of-Artificial-Intelligence-in-future-technology.pdf (researchgate.net)

[3] SHINDE, P. P. and SHAH, S. A Review of Machine Learning and Deep Learning Applications, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6. DOI: 10.1109/ICCUBEA.2018.8697857.

[4] CARVALHO, A., JUSTO, J.F., ANGELICO, B.A. et al. Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum, Applied Soft Computing, 2022, 109927, ISSN 1568-4946. DOI: 10.1016/j.asoc.2022.109927.

[5] CARVALHO, A. Função de Ativação, o Núcleo da Composição de Neurônios Artificiais, EAILAB, IFSP, 2024. Disponível em: https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-artificiais/

[6] CARVALHO, A., JUSTO, J.F., ANGELICO, B.A. et al., Rotary Inverted Pendulum Identification for Control by Paraconsistent Neural Network, in IEEE Access, 2021. DOI: 10.1109/ACCESS.2021.3080176.

[7] MONTAZER, G. A. et al. Radial basis function neural networks: A review. Computer Reviews. Journal, v. 1, n. 1, p. 52-74, 2018. Disponivel em: https://www.ise.ncsu.edu/fuzzy-neural/wp-content/uploads/sites/9/2022/08/RBFNN.pdf

[8] DEEP LEARNING BOOK. Capítulo 51 – Arquitetura de Redes Neurais Long Short Term Memory (LSTM). Disponível em: https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-long-short-term-memory/. Acessado em Abr 03, 2024.

[9] OLAH, C. Understanding LSTM Networks, Colah's blog. Aug 2015. Disponível em: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Acessado em Abr 03, 2024.

[10] YEOLA, C. Convolutional Neural Network (CNN) In Deep Learning, Feb, 2022. Disponível em: https://python.plainenglish.io/convolution-neural-network-cnn-in-deep-learning-77f5ab457166. Acessado em Abr 03, 2024.

[11] EDGE IMPULSE. Build, train, optimize, ai for the edge. Edge Impulse. Disponível em: https://edgeimpulse.com. Acessado em abr 03, 2024.

[12] CARVALHO, A. EAILAB inicia atividades com produção intensa de trabalhos. Nov, 2023. Disponível em: https://eailab.labmax.org/2023/11/16/eailab-inicia-atividades-com-producao-intensa-de-trabalhos/. Acessado em Abr 03, 2024.

[13] CHIEN, J-T. Chapter 7 – Deep Neural Network, Editor(s): Jen-Tzung Chien, Source Separation and Machine Learning, Academic Press, 2019, 259-320 p, ISBN 9780128177969, DOI: 10.1016/B978-0-12-804566-4.00019-X. Disponível em: https://www.sciencedirect.com/science/article/pii/B978012804566400019X.

[14] NIELSEN, M. A. Neural networks and deep learning. San Francisco, CA, USA: Determination press, 2015.

[15] SHINDE, Pramila P.; SHAH, Seema. A review of machine learning and deep learning applications. In: 2018 Fourth international conference on computing communication control and automation (ICCUBEA). IEEE, 2018. p. 1-6. DOI: 10.1109/ICCUBEA.2018.869.

Elaborado Por: Dr. Arnaldo de Carvalho Junior
Publicado em: Abr 03, 2024
Disponível em:
https://eailab.labmax.org/2024/04/03/redes-neurais-artificiais-algoritmos-poderosos-para-aplicacoes-de-ia-e-ml/. Acessado em Nov 08, 2024.

Post 6: Datasets de Acesso Livre para Projetos de IA!

A melhor forma de aprender o poder da inteligência artificial (IA) é criando projetos e aplicações com ela. O sucesso de um projeto de IA começa pela análise, seleção, tratamento e filtragem do conjunto de dados a serem utilizados no treinamento da IA. O tempo demandado nesta etapa permitirá economizar tempo nas etapas futuras do projeto, minimizar o uso de recursos computacionais e elevar o nível de acurácia da IA.

A seguir são relacionadas alguns dos repositórios de conjuntos de dados (*datasets*) de acesso livre para projetos de IA mais difundidos.

1. Datasets de Universidades

a) UCI Machine Learning Repository – https://archive.ics.uci.edu/datasets

b) Harvard Dataverse (Harvard University) – https://data.harvard.edu/dataverse

c) Labelme (CSAIL – MIT)
http://labelme.csail.mit.edu/Release3.0/browserTools/php/dataset.php–

d) Center of AI in Medicine & Imaging (Stanford University)
– https://aimi.stanford.edu/shared-datasets

2. Datasets de Governos

a) Data .gov US – https://data.gov/

b) Data .gov UK – https://www.data.gov.uk/

c) European Data – https://data.europa.eu/data/datasets?locale=en

d) Latin American Data Bank – https://ropercenter.cornell.edu/latin-american-data-bank

e) Dados Abertos Brasil – https://dados.gov.br/signin

3. Datasets de Astronomia e Espaço

a) Earth Data (NASA) – https://www.earthdata.nasa.gov/

b) CERN Open Data Portal – https://opendata.cern.ch/

4. Datasets de Saúde

a) Health Data (USA) – https://healthdata.gov/

b) Centers For Disease Control And Prevention (USA)

– https://www.cdc.gov/datastatistics/index.html

c) Dataset for Health Care and Public Health (USA)

– https://researchguides.dartmouth.edu/c.php?g=517073&p=6289098

d) Global Health Observatory Data Repository – World Health Organization (WHO) – https://apps.who.int/gho/data/node.home

e) National Library of Medicine (NIH – USA) – https://medpix.nlm.nih.gov/home

5. Datasets De Tópicos Variados

a) ImageNet – https://image-net.org/

b) Kaggle Datasets – https://www.kaggle.com/datasets

c) Sigma Open Dadasets – https://sigma.ai/open-datasets/

d) OpenML – https://www.openml.org/

e) Datahub .io – https://datahub.io/collections

f) FiveThirtyEight – https://data.fivethirtyeight.com/

g) Non-Commercial Datasets – https://developer.imdb.com/non-commercial-datasets/

h) Google Dataset Search – https://datasetsearch.research.google.com/

i) IBM Data Asset eXchange – https://developer.ibm.com/exchanges/data/

j) AWS Open Data – https://aws.amazon.com/marketplace/search/results?trk=868d8747-614e-4d4d-9fb6-fd5ac02947a8&sc_channel=el&FULFILLMENT_OPTION_TYPE=DATA_EXCHANGE&CONTRACT_TYPE=OPEN_DATA_LICENSES&filters=FULFILLMENT_OPTION_TYPE%2CCONTRACT_TYPE

BD – https://basedosdados.org/

6. Datasets Temáticos

a) Furnas Dataset (Electrical Power Transmission Lines) – https://github.com/freds0/PTL-AI_Furnas_Dataset?tab=readme-ov-file

b) Nasdaq Data Link – https://data.nasdaq.com/institutional-investors

c) Antarctic Datasets – https://www.antarcticglaciers.org/antarctica-2/antarctic-datasets/

d) BFI film industry statistics (UK) – https://www.bfi.org.uk/industry-data-insights

e) NYC Taxi Trip Data – https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

f) FBI (USA) Crime Data Explorer – https://cde.ucr.cjis.gov/LATEST/webapp/#/pages/home

Elaborado Por: Dr. Arnaldo de Carvalho Junior

Publicado em: Jun 17, 2024

Disponível em: https://eailab.labmax.org/2024/06/17/datasets-de-acesso-livre-para-projetos-de-i-a/. Acessado em Nov 08, 2024.

Post 7: Excelentes Recursos para Estudar Aprendizado de Máquina

Neste post são apresentados excelentes recursos gratuitos para aprendizado de inteligência artificial (IA) e aprendizado de máquina (*machine learning* – ML) [1][3].

1. CURSOS, GUIAS E TUTORIAIS

21. **TensorFlow:** tutoriais oficiais – https://www.tensorflow.org/tutorials?hl=pt-br
22. **Scikit-learn:** Guias compreensivos e exemplos – https://scikit-learn.org/stable/user_guide.html
23. **Kaggle:** trilhas para estudar aprendizado de máquina (*machine learning* – ML), Python, Panda, etc. – https://www.kaggle.com/learn
24. **Google AI:** como aprender a plataforma de inteligência artificial – IA (*artificial intelligence* – AI) do Google – https://ai.google/build
25. **FreeCodeCamp:** curso de Python com ML – https://www.freecodecamp.org/learn/machine-learning-with-python/
26. **Stanford CS224N:** Aprendizado profundo (*deep learning* – DL) para processamento de linguagem natural (*natural language processing* NLP) – https://web.stanford.edu/class/cs224n/
27. **MIT OpenCourseWare:** introdução à DL – https://ocw.mit.edu/courses/6-s191-introduction-to-deep-learning-january-iap-2020/
28. **TinyMLEdu:** iniciativa aberta de treinamento em TinyML, com plataforma Edge Impulse – https://tinyml.seas.harvard.edu/
29. **Deep Learning Book:** plataforma online, em português, abordando conceitos de redes neurais, IA, ML, DL, em constante evolução [2] – https://www.deeplearningbook.com.br/
30. **Fast.ai:** DL prático para programadores – https://course.fast.ai/
31. **DataCamp:** tutoriais gratuitos de Python e R – https://www.datacamp.com/tutorial
32. **Canais do Youtube:**

33. **Sentdex:** tutoriais Python e ML – https://www.youtube.com/user/sentdex

34. **Corey Schafer:** Programação Python – https://www.youtube.com/user/schafer5

2. CURSOS DE IA NVIDIA

Vale a pena revisitar o post anterior do EAILab, indicado na referência [3], onde são apresentados 8 cursos gratuitos disponibilizados pela NVIDIA. Alguns dos cursos requerem o uso dos módulos e placas de computador único da empresa.

3. LIVROS IA

O EAILab em sua seção “HotLinks” [4] mantém uma lista atualizada de livros de acesso gratuito para aprofundar os conhecimentos em inteligência artificial, aprendizado de máquina, programação Python, algoritmos de raciocínio com lógicas não classicas (Paraconsistente, Fuzzy, …), entre outros.

4. REFERÊNCIAS

[1] ALAM A. The Best Free Resources to Master Machine Learning. Art of Data Science, 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQHbF_FxS0qniQ/feedshare-shrink_800/0/1720438696292?e=1723075200&v=beta&t=4RuIg0quCep9a4wOBjMtPHPqxV4DI-AN58jGc_E02Js. Acessado em Julho 08, 2024.

[2] DSA. Deep Learning Book, Data Science Academy (DSA), 2024. Disponível em: <https://www.deeplearningbook.com.br/>. Acessado em Julho 8, 2024.

[3] CARVALHO JUNIOR, A. 8 Excelentes cursos gratuitos da NVIDIA, para entrar no universo da Inteligência Artificial. EAILab Posts, Abril 2, 2024. Disponível em: https://eailab.labmax.org/2024/04/02/8-excelentes-cursos-gratuitos-da-nvidia-para-entrar-no-universo-da-inteligencia-artificial/. Acessado em Jul 8, 2024.

[4] CARVALHO JUNIOR, A. Hot Links – Free Ebooks, EAILab, 2024. Disponível em: <https://eailab.labmax.org/hot-links/>. Acessado em Julho 8, 2024.

Preparado por: Dr. Arnaldo de Carvalho Junior

Publicado em: Jul 08, 2024

Disponível em: https://eailab.labmax.org/2024/07/08/excelentes-recursos-para-estudar-aprendizado-de-maquina/. Acessado em Nov 08, 2024.

Post 8: Principais Algoritmos Utilizados em Inteligência Artificial

1. INTRODUÇÃO

Conforme apresentado em postagens anteriores [1],[2] a Inteligência Artificial (IA) é uma área da ciência da computação que desenvolve sistemas capazes de executar tarefas normalmente realizadas por seres inteligentes.

A IA tem sido utilizada nas mais diversas áreas do conhecimento humano, para raciocinar, identificar, classificar, descobrir significado, generalizar, aprender por experiência, tomar decisão, controlar e realizar predições ou prognósticos [1].

2. PRINCIPAIS ALGORITMOS DE IA

Diferentes técnicas matemáticas, estatísticas e computacionais são utilizadas para o desenvolvimento de soluções de IA, conforme a Figura 1. Um verdadeiro arsenal, ou "cinto de utilidades", de diferentes recursos estão à disposição do pesquisador em IA. Esses algoritmos podem ser utilizados de forma isolada ou combinados, em soluções híbridas, para uma melhor performance.

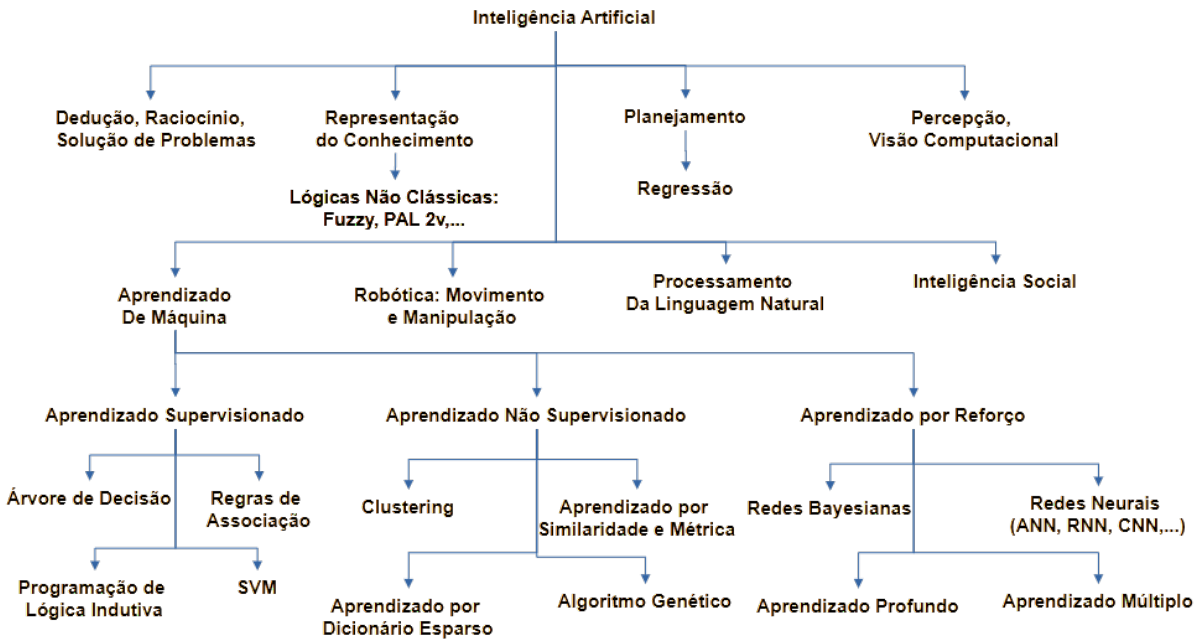


Figura 1 – Algoritmos de IA e ML
Fonte: O Autor (2024)

A seguir são relacionados os principais algoritmos utilizados em IA [3],[4]. Para saber mais, explore as referências indicadas em cada um dos algoritmos.

2.1. Regressão Linear (*Linear Regression* – LR): Um dos mais conhecidos algoritmos de estatística e muito utilizado no aprendizado de máquinas (*machine learning* – ML). A LR cria um modelo para a compreensão da relação entre variáveis numéricas de entrada e saída. Isso permite que a LR seja utilizada para prever resultados com base em dados anteriores [5].

2.2. Regressão Logística (*Logistic Regression* – LR): É um modelo estatístico utilizado que analisa a relação entre 2 fatores de dados. A regressão logística é um algoritmo de ML supervisionado, usado para tarefas de classificação, cujo objetivo é prever a probabilidade de uma instância pertencer ou não a uma determinada classe. É o método estatístico preferido para problemas de classificação binária (problemas com dois valores de classe).[6].

2.3. Aumento Gradual (*Gradient Boosting* – GB): É uma técnica em melhora predições através de erros passados em passos pequenos. O GB é uma das técnicas mais poderosas para a construção de modelos preditivos [7].

2.4. Bayes Ingênuos (*Naive Bayes* – NB): É um algoritmo simples porém poderoso para modelagem preditiva. O Teorema de Bayes fornece uma maneira de calcular a probabilidade de uma hipótese dado o conhecimento prévio [8].

2.5. Redes Bayesianas (*Bayesian Networks* – BN): Também fundamentado no Teorema de Bayes, utiliza probabilidade para realizar predição considerando diferentes fatores. as Redes Bayesianas fornecem uma ferramenta útil para visualizar o modelo probabilístico para um domínio, revisar todas as relações entre as variáveis aleatórias, e razão sobre probabilidades causais para cenários dadas as evidências disponíveis [9].

2.6. Modelo de Markov (Markov *Model* – MM): São uma classe de Modelos Gráficos Probabilísticos (*probabilistic graph models* – PGM) que representam processos dinâmicos, ou seja, um processo que não é estático, mas sim que muda com o tempo. Em particular, preocupa-se mais sobre como o estado (*state*) de um processo muda com o tempo [8]. Em

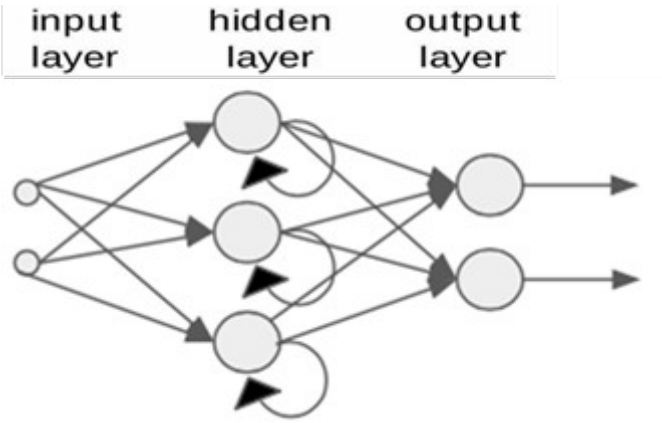
conjunto com a amostragem de Monte Carlo, a cadeia de Markov fornece uma classe de algoritmos para amostragem aleatória sistemática a partir de distribuições de probabilidade de alta dimensão. Ao contrário dos métodos de amostragem de Monte Carlo que são capazes de extrair amostras independentes da distribuição, os métodos de Monte Carlo da Cadeia de Markov coletam amostras onde a próxima amostra depende da amostra existente, chamada de Cadeia de Markov. Isso permite que os algoritmos se restrinjam na quantidade que está sendo aproximada da distribuição, mesmo com um grande número de variáveis aleatórias [10].

2.7. Árvores de Decisão (*Decision Tree* – DT): é um algoritmo útil de aprendizado de máquina usado para tarefas de regressão e classificação. O nome “Árvore de Decisão” vem do fato de que o algoritmo continua dividindo o conjunto de dados em porções cada vez menores até que os dados sejam divididos em instâncias únicas, que são classificadas, ao tomar decisões do tipo sim ou não em cada instância. Ao visualizar a estrutura dos resultados do algoritmo, a maneira como as categorias são divididas se parecerem com uma árvore e muitas folhas [12],[13].

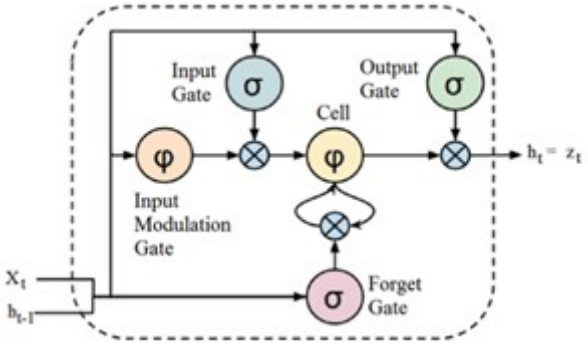
2.8. Raciocínio por Inferência (*Reasoning by Inferences*): Lógicas não clássicas como lógica difusa (Fuzzy) [14], paraconsistente e suas variações como a lógica paraconsistente anotada de 2 valores (PAL2v) [15],[16],[17],[18], permitem a construção de algoritmos para tomada de decisão quando informações não estão claramente definidas. Algoritmos PAL2v de referência estão disponíveis em https://sites.google.com/view/prof-arnaldo/pal2v-key-points?authuser=0.

2.9. Redes Neurais Artificiais (A*rtificial Neural Networks* – ANN): redes de neurônios artificiais baseados no cérebro biológico, que aprende por exemplos [1]. Cada neurônio é bloco elementar da rede neural, formado por um conjunto de entradas e saídas que funcionam como sinapses da ANN [5],[6],[7]. O sinal que atravessa uma sinapse pode ter um peso (reforço) maior ou menor ao entrar em um neurônio, dependendo do aprendizado da rede neural. As entradas de cada neurônio são combinadas e aplicadas a uma função de ativação, que processará e apresentará o resultado em sua saída [2],[3]. As ANNs já foram abordadas em postagem anterior [1].

2.10. Redes Neurais Recorrentes (*Recurrent Neural Networks* – RNN): tipo de rede neural, cuja arquitetura possui realimentações internas, funcionando como efeito memória, permitindo que a RNN entenda sequências como texto e séries temporais de dados [1]. As RNAs podem assim, serem utilizadas na predição e controle de sistemas [6],[7]. Uma evolução das RNAs são as redes de Memória de Termo Curto Longo (*long short term memory* – LSTM) e Unidades de Portas Recorrentes (*gated recurrent units* – GRU). As redes LSTM e GRU já foram abordadas em postagem anterior [1].



(a)



(b)

Figura 2 – Conceito de RNN (a) e Unidade LSTM (b).
Fonte: Adaptado de [1].

2.11. Redes Neurais Convolucionais (*Convolutional Neural Networks* – CNN): são redes neurais especializadas, que permitem computadores ver e entender imagens [1],[2], [3],[19]. Neste tipo de ANN, **técnicas de convolução** são aplicadas na matriz de dados (como a matriz de pixels de uma imagem), antes da entrada na ANN, de modo a destacar ou realçar determinadas características como objeto, cores, contraste, contornos, etc. São muito utilizadas na identificação e classificação de padrões de imagens. Um exemplo de arquitetura pode ser visto na Figura 3.

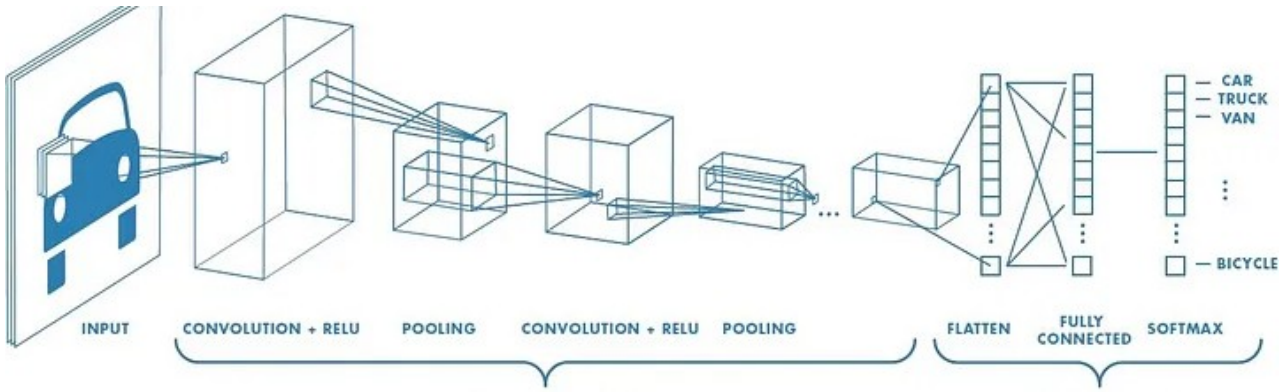


Figura 3 – Arquitetura de Rede Neural Convolucional.
Fonte: Adaptado de [20].

2.12. Conjunto de Significados K (*K-Means Clustering*): algoritmo que agrupa itens semelhantes, sem uma definição prévia. É frequentemente usado como uma técnica de análise de dados para descobrir padrões interessantes em dados, como grupos de clientes com base em seu comportamento [21].

2.13. Análise de Componentes Principais (*Principal Component Analysis* – PCA): A PCA é uma técnica de aprendizado de máquina não supervisionada que empacota dados importantes em um pequeno espaço. É usada principalmente na análise de componentes principais seja a redução da dimensionalidade. Além de usar o PCA como técnica de preparação de dados, também pode-se utilizá-la para ajudar a visualizar dados. Com os dados visualizados, é mais fácil obter algumas ideias e decidir sobre a próxima etapa em modelos de aprendizado de máquina [22].

2.14. Auto-Codificadores (*AutoEncoders*): Autoencoders (AE) são redes neurais que visam copiar suas entradas para suas saídas. Especificamente, projetaremos uma arquitetura de rede neural de modo a impor um gargalo na rede que força uma representação de conhecimento compactada da entrada original. Se houver algum tipo de estrutura nos dados (ou seja, correlações entre os recursos de entrada), essa estrutura poderá ser aprendida e consequentemente aproveitada ao forçar a entrada através do gargalo da rede. Assim, o AE comprime e depois reconstrói imagens [23].

2.15. Aprendizagem de Reforço (*Reinforcement Learning* – RL): aprende com recompensas, ou seja, o computador é recompensado por boas ações e erros são corrigidos [24].

2.16. Aprendizado Q (*Q-Learning*): encontra o melhor caminho em um labirinto, através da exploração e recompensa. O processo de Q-Learning cria uma matriz (tabela) exata para o agente a qual ele “consulta” para maximizar sua recompensa a longo prazo durante seu aprendizado [25].

2.17. Vizinhos Mais Próximos (*K Nearest Neighbors* – K-NN): encontra vizinhos mais próximos para fazer predições. As previsões são feitas para uma nova instância (x) pesquisando todo o conjunto de treinamento para as instâncias mais semelhantes (os vizinhos) e resumindo a variável de saída para essas instâncias K. Para regressão, essa pode ser a variável de saída média, na classificação, esse pode ser o valor do modo (ou mais comum) [26].

2.18. Florestas Aleatórias (*Randon Forest* – RF): combina várias respostas para precisão. O RF utiliza um conjunto de Árvores de Decisão para realizar tarefas de classificação ou regressão. O método da Floresta Aleatória, que se dá pela combinação de outros métodos são denominados como ensemble. O conceito fundamental das Florestas Aleatórias é a sabedoria das multidões: um número grande de modelos não correlacionados, no caso as árvores, operando em conjunto performarão melhor do que cada modelo individual [27].

2.19. Máquina Vetorial de Suporte (S*upport Vector Machine* – *S*VM): Máquinas de vetores de suporte são um conjunto de métodos de aprendizado supervisionado utilizados para classificação, regressão, e detecção de outliers. Todas essas são tarefas comuns em aprendizado de máquina. Um classificador SVM linear simples funciona criando uma linha reta entre duas classes. Isso significa que todos os pontos de dados de um lado da linha representarão uma categoria, e os pontos de dados do outro lado da linha serão colocados em uma categoria diferente. Isso significa que pode haver um número infinito de linhas para escolher [28].

2.20. Algoritmos Genéticos (*Genetic Algorithm* – GA): O GA é um metaheurístico inspirado no processo de seleção natural que pertence à maior classe de algoritmos evolutivos (EA). Ele evolui na solução ao combinar as melhores opções através do tempo. Os algoritmos genéticos são comumente usados para gerar soluções de alta qualidade para otimização e problemas de pesquisa, confiando em operadores biologicamente inspirados,

como mutação, c*rossover* e seleção. Exemplos de aplicações de GA incluem otimizar as árvores de decisão para melhor desempenho, resolver quebra-cabeças otimização de hiperparâmetro, inferência causal, etc. [29].

2.21. Redes Transformadoras (*Transformer Networks* – TN): As redes Transformer vem ganhando bastante interesse entre pesquisadores nos últimos anos. O Transformer é um modelo de aprendizado profundo introduzido em 2017 que utiliza o mecanismo de atenção, pesando a influência de diferentes partes dos dados de entrada. O Transformer é uma arquitetura que visa resolver tarefas sequência-à-sequência enquanto lida com dependências de longo alcance com facilidade. Ele se baseia inteiramente na auto atenção (*Self-Attention*) para computar as representações de sua entrada e saída sem usar RNNs (Redes Neurais Recorrentes) alinhadas em sequência ou convolução. São algoritmos complexos e que exigem grande capacidade computacional. As redes Transformer tem sido utilizadas no processamento de linguagem natural e análise de séries temporais [30]. A Figura 4 apresenta um modelo da arquitetura de Rede Transformer [31].

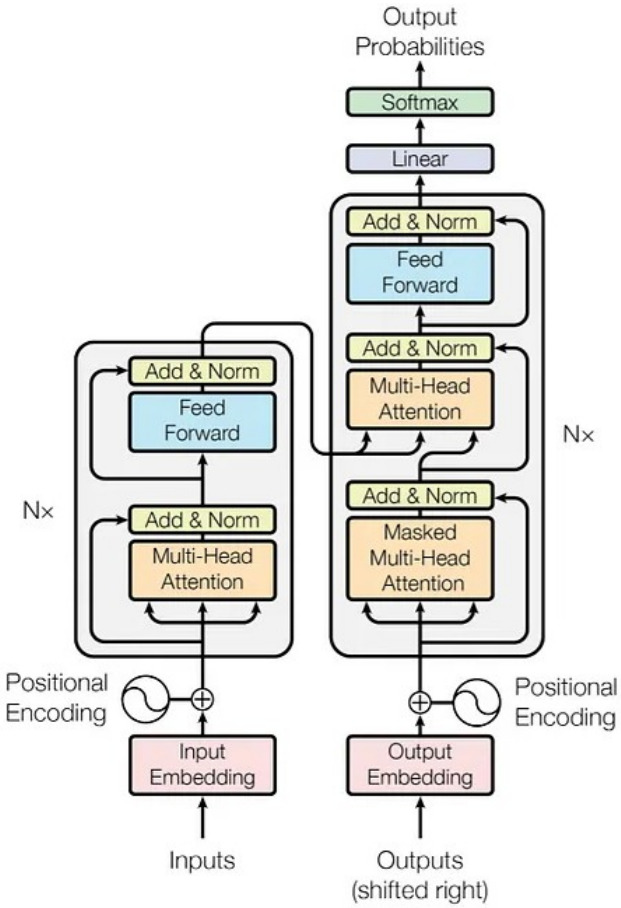


Figura 4 – Modelo de Arquitetura do Algoritmo de Rede Transformer
Fonte: Adaptado de [31]

3. CONCLUSÃO

Esta postagem procurou agrupar e relacionar as principais classes de algoritmos de IA e ML. Para cada algoritmo relacionado, há inúmeras variações e adaptações. Como pôde ser verificado, há dezenas de técnicas, cada uma apresenta vantagens, desvantagens e principais usos. É importante que o pesquisador de IA conheça fundamentos de matemática, como álgebra linear, probabilidade e estatística e cálculo, bem como programação de modo a avaliar a melhor ferramenta de IA para cada desafio que se deseja resolver. Há inúmeras bibliotecas disponíveis para Python, para a maioria dos algoritmos de IA, o que pode acelerar o desenvolvimento de novas ferramentas e aplicações.

4. REFERÊNCIAS

[1] CARVALHO, Arnaldo. Redes Neurais Artificiais: Algoritmos poderosos para aplicações de IA e ML. EAILAB, IFSP, Publicado em Abril 03, 2024. Disponível em <https://eailab.labmax.org/2024/04/03/redes-neurais-artificiais-algoritmos-poderosos-para-aplicacoes-de-ia-e-ml/>. Acessado em Jun 20, 2024.

[2] CARVALHO, Arnaldo. 10 Tendências de Aplicação de Inteligência Artificial em 2024! EAILAB, IFSP, Publicado em Nov 28, 2023. Disponível em <https://eailab.labmax.org/2023/11/28/10-tendencias-de-aplicacao-de-inteligencia-artificial-em-2024/>. Acessado em Jun 20, 2024.

[3] RAY, Susmita. A quick review of machine learning algorithms. In: 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, 2019. p. 35-39.

[4] MAHESH, Batta. Machine learning algorithms- A Review. International Journal of Science and Research (IJSR).[Internet], v. 9, n. 1, p. 381-386, 2020.

[5] BROWNLEE, J. Linear Regression for Machine Learning, Mahine Learning Mastery, Dez 6, 2023. Disponível em <https://machinelearningmastery.com/linear-regression-for-machine-learning/>. Acessado em Jul 7, 2024.

[6] BROWNLEE, J. Logistic Regression for Machine Learning, Mahine Learning Mastery, Dez 6, 2023. Disponível em <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>. Acessado em Jul 7, 2024.

[7] BROWNLEE, J. A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning, Ago 15, 2020. Disponível em <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>. Acessado em Jul 7, 2024.

[8] BROWNLEE, J. Naive Bayes for Machine Learning, Ago 15, 2020. Disponível em <https://machinelearningmastery.com/naive-bayes-for-machine-learning/>. Acessado em Jul 7, 2024.

[9] BROWNLEE, J. A Gentle Introduction to Bayesian Belief Networks, Set 25, 2019. Disponível em <https://machinelearningmastery.com/introduction-to-bayesian-belief-networks/>. Acessado em Jul 7, 2024.

[10] LAWHATRE, P. Gentle Introduction to Markov Chain, Machine Learnig Plus, 2024. Disponível em: <https://www.machinelearningplus.com/markov-chain/>. Acessado em Jul 7, 2024.

[11] BROWNLEE, J. A Gentle Introduction to Markov Chain Monte Carlo for Probability, Set 25, 2019. Disponível em <https://machinelearningmastery.com/markov-chain-monte-carlo-for-probability//>. Acessado em Jul 7, 2024.

[12] CHARBUTY, Bahzad; ABDULAZEEZ, Adnan. Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, v. 2, n. 01, p. 20-28, 2021.

[13] SOMVANSHI, Madan *et al.* A review of machine learning techniques using decision tree and support vector machine. In: 2016 international conference on computing communication control and automation (ICCUBEA). IEEE, 2016. p. 1-7.

[14] HÜLLERMEIER, Eyke. Fuzzy methods in machine learning and data mining: Status and prospects. Fuzzy sets and Systems, v. 156, n. 3, p. 387-406, 2005.

[15] CARVALHO, Arnaldo. Função de Ativação, o Núcleo da Composição de Neurônios Artificiais, EAILAB, IFSP, 2024. Disponível em: <https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-artificiais/>. Acessado em Jun 20, 2024.

[16] CARVALHO, A., JUSTO, J.F., ANGELICO, B.A. *et al*. Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum, Applied Soft Computing, 2022, 109927, ISSN 1568-4946. DOI: 10.1016/j.asoc.2022.109927.

[17] CARVALHO, A., JUSTO, J.F., ANGELICO, B.A. *et al*., Rotary Inverted Pendulum Identification for Control by Paraconsistent Neural Network, in IEEE Access, 2021. DOI: 10.1109/ACCESS.2021.3080176.

[18] DE CARVALHO JUNIOR, Arnaldo *et al.* A comprehensive review on paraconsistent annotated evidential logic: Algorithms, Applications, and Perspectives. Engineering Applications of Artificial Intelligence, v. 127, p. 107342, 2024.

[19] GUPTA, Jaya; PATHAK, Sunil; KUMAR, Gireesh. Deep learning (CNN) and transfer learning: a review. In: Journal of Physics: Conference Series. IOP Publishing, 2022. p. 012029.

[20] SAHA, S. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, Medium, Dez 2018. Disponível em <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Acessdo em Jul 12, 2024.

[21] BROWNLEE, J. 10 Clustering Algorithms With Python, Mahine Learning Mastery, Ago 20, 2020. Disponível em <https://machinelearningmastery.com/clustering-algorithms-with-python/>. Acessado em Jul 7, 2024.

[22] TAM, A. 10 Principal Component Analysis for Visualization, Mahine Learning Mastery, Out 27, 2021. Disponível em <https://machinelearningmastery.com/principal-component-analysis-for-visualization/>. Acessado em Jul 7, 2024.

[23] DEEP LEARNING BOOK, Capítulo 58 – Introdução aos Autoencoders, Data Science Academy, 2022. Disponível em <https://www.deeplearningbook.com.br/introducao-aos-autoencoders/>. Acessado em Jul 12, 2024.

[24] DEEP LEARNING BOOK, Capítulo 62 – O Que é Aprendizagem Por Reforço?, Data Science Academy, 2022. Disponível em <https://www.deeplearningbook.com.br/o-que-e-aprendizagem-por-reforco/>. Acessado em Jul 12, 2024.

[25] DEEP LEARNING BOOK, Capítulo 70 – Deep Q-Network e Processos de Decisão de Markov, Data Science Academy, 2022. Disponível em <https://www.deeplearningbook.com.br/deep-q-network-e-processos-de-decisao-de-markov/>. Acessado em Jul 12, 2024.

[26] BROWNLEE, J. K-Nearest Neighbors for Machine Learning, Ago 15, 2020. Disponível em <https://machinelearningmastery.com/markov-chain-monte-carlo-for-probability//>. Acessado em Jul 7, 2024.

[27] KUBRUSLY, J. Introdução ao Machine Learning – Capítulo 4 Floresta Aleatória, Laboratório de Estatística, Universidade Federal Fluminense, 2023. Disponível em: <https://bookdown.org/jessicakubrusly/intr-machine-learning-i/_book/cap-floresta.html>. Acessado em Jul 12, 2024.

[28] SOUZA, I. C. N. Tutorial de aprendizado de máquina SVM – o que é o algoritmo de máquina de vetores de suporte, explicado com exemplos de códigos. FreeCodeCamp, Jun 2024. Disponível em: <https://www.freecodecamp.org/portuguese/news/tutorial-de-aprendizado-de-maquina-svm/>. Acessado em Jul 12, 2024.

[29] BROWNLEE, J. Simple Genetic Algorithm From Scratch in Python, Mahine Learning Mastery, Out 12, 2021. Disponível em <https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/>. Acessado em Jul 7, 2024.

[30] DEEP LEARNING BOOK, Capítulo 85 – Transformadores – O Estado da Arte em Processamento de Linguagem Natural, Data Science Academy, 2022. Disponível em <https://www.deeplearningbook.com.br/transformadores-o-estado-da-arte-em-processamento-de-linguagem-natural/>. Acessado em Jul 12, 2024.

[31] ANKIT, U. Transformer Neural Network: Step-By-Step Breakdown of the Beast. Toward Data Science, Medium, Abr 2020. Disponível em: <https://towardsdatascience.com/transformer-neural-network-step-by-step-breakdown-of-the-beast-b3e096dc857f>. Acessado em Jul 12, 2024.

Elaborado Por: Dr. Arnaldo de Carvalho Junior

Publicado em: Jul 13, 2024

Disponível em: https://eailab.labmax.org/2024/07/08/excelentes-recursos-para-estudar-aprendizado-de-maquina/. Acessdo em Nov 08, 2024.

Post 9: O Poder Das CNNs Em Aplicações de ML Envolvendo Identificação e Classificação de Imagens

1.O QUE É UMA CNN?

Uma rede neural convolucional (convolutional neural network – CNN) é um tipo especializado de algoritmo de aprendizado profundo (deep learning) projetado para processar e analisar dados visuais. É muito utilizada em aplicações de aprendizado de máquina (machine learning – ML) [1]. Inspirados no córtex visual humano, as CNNs usam operações de convolução para extrair recursos e identificar padrões dentro das imagens. A CNN é um tipo de rede neural do tipo "olhar para a frente" (feedforward neural network – FNN) que aprende uma característica por meio de otimização de filtros [1]. Essa arquitetura permite que as CNNs interpretem e classifiquem imagens com eficiência, tornando-as inestimáveis em aplicativos de visão computacional [2]. A Figura 1 apresenta uma arquitetura de uma CNN.

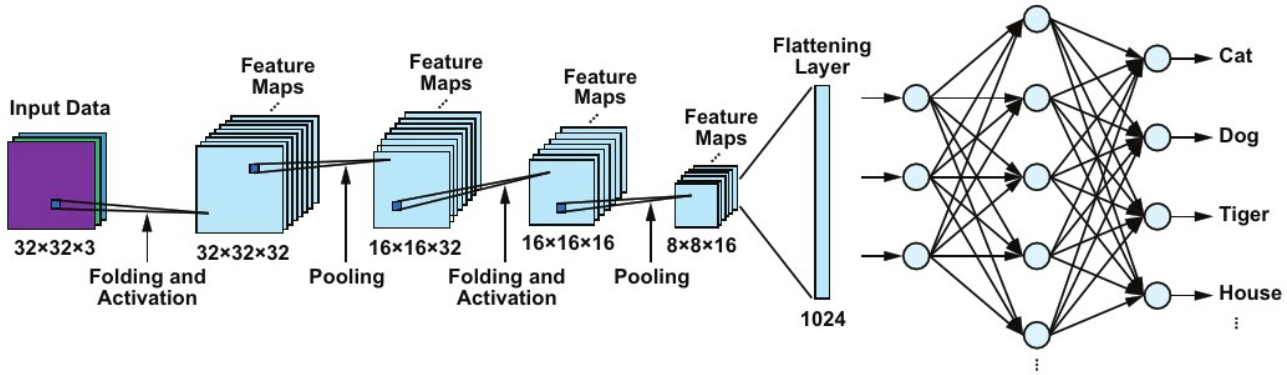


Figura 1 – Arquitetura de CNN
Fonte: Adaptado de [2].

2. CAMADAS-CHAVE DA CNN

a) Camada convolucional (convolutional layer): este é o bloco de construção principal de uma CNN, onde os filtros (kernels) deslizam sobre a imagem de entrada para detectar recursos (features) como bordas, contrastes e texturas. A saída é um mapa de recursos que destaca esses recursos detectados. A Figura 2 apresenta as atividades da camada convolucional. Já a sequência de imagens da Figura 3 apresenta um exemplo do processo convolucional. Vários filtros de convolução podem ser aplicados,

procurando-se extrair características específicas de um padrão 2D como contorno, contraste, objeto, etc [1].

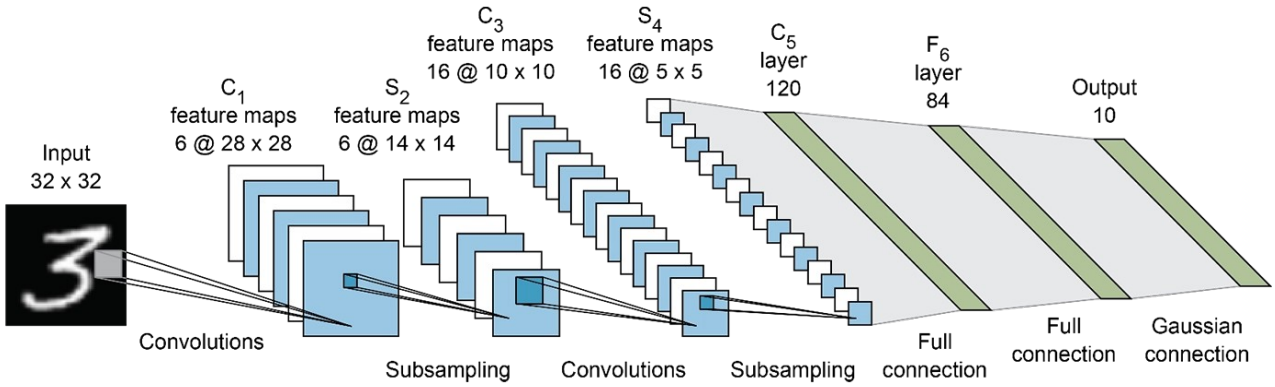


Figura 2 – Série de Convoluções como Forma de Extração de Características da Imagem, pela CNN.
Fonte: Adaptado de [3].

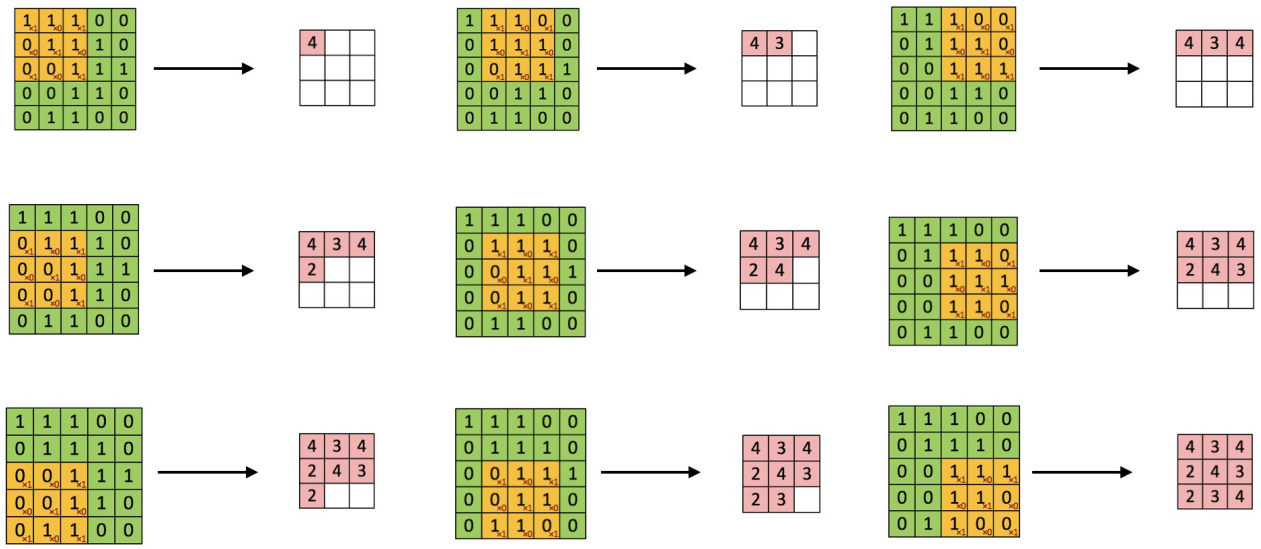
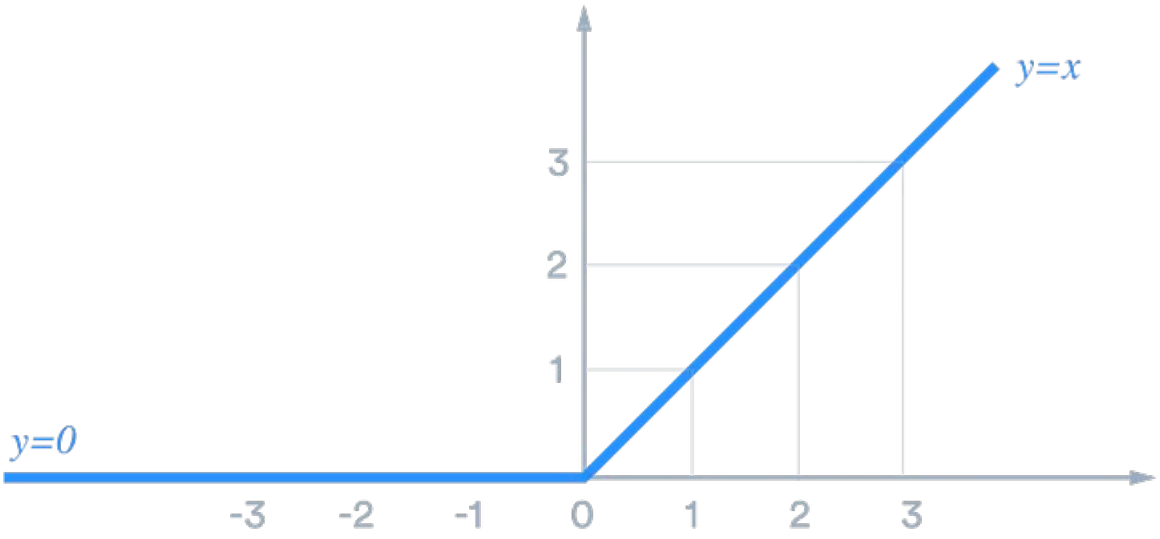


Figura 3 - Sequência de Convolução.
Fonte: Adaptado de [4].

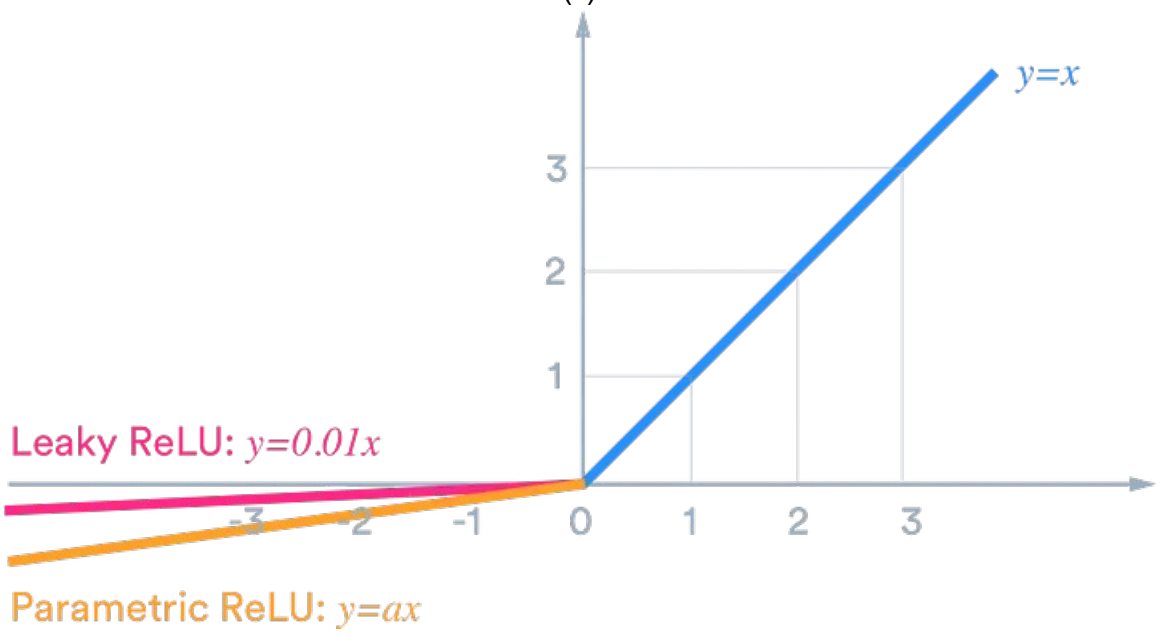
b) Camada de agrupamento (pooling layer): Após a camada convolucional, as camadas de agrupamento diminuem os mapas de recursos, reduzindo suas dimensões espaciais. Esta etapa diminui a complexidade computacional e ajuda a evitar o excesso de ajuste. O máximo de pool, que seleciona o valor máximo de uma região, é uma técnica comum usada aqui.

c) Camada de ativação da unidade retificadora linear (rectified linear unit – ReLU): A ReLU introduz a não linearidade no modelo, permitindo que ele aprenda padrões e relacionamentos complexos nos dados. A ReLU é uma das funções de ativação mais utilizadas atualmente em redes neurais pela sua rapidez no

processo de aprendizagem [5]. Na função ReLU, se a entrada for negativa, ela será convertida em zero e o neurônio não será ativado. Isso significa que, ao mesmo tempo, apenas alguns neurônios são ativados, tornando a rede esparsa, eficiente e de computação fácil. Essa vantagem também pode ser considerada uma desvantagem, pois os neurônios utilizando ReLU tendem a "morrer" durante o treinamento, causando a saída do neurônio iniciar a produzir apenas zeros [6]. Para evitar isso, uma variação da ReLU, chamada Leaky-ReLU (LReLU) foi desenvolvida, que aplica uma ligeira inclinação na função de ativação para valores menores que zero [7]. A Figura 4 apresenta as funções de ativação ReLU e sua variante Leaky-ReLU [8].



(a)



(b)

Figura 4 – Funções de Ativação ReLU e Leaky-ReLU.
Fonte: Adaptado de [8].

d) Camada totalmente conectada (fully connected layer): Nos estágios finais, as camadas totalmente conectadas usam os recursos de alto nível aprendidos pelas camadas anteriores para fazer previsões. Cada neurônio nessas camadas está conectado a todos os neurônios da camada anterior, integrando todos os recursos extraídos para produzir a saída final.

3. CNN X ANN

Embora as redes neurais artificiais (artificial neural networks – ANNs) sejam versáteis [1], elas lutam com dados de alta dimensão, como imagens devido à sua natureza totalmente conectada. As CNNs, no entanto, são projetadas especificamente para lidar com estruturas de dados semelhantes à grade (grid) ou matriz, tornando-as altamente eficientes para o processamento de imagens. Sua capacidade de capturar hierarquias espaciais e padrões locais os diferencia das ANNs tradicionais.

3.1. A ANN NÃO SERVE PARA PROCESSAR IMAGENS?

As ANNs exigem engenharia extensa de recursos e lutam com a alta dimensionalidade dos dados da imagem. Por outro lado, as CNNs aprendem automaticamente representações hierárquicas de recursos, melhorando significativamente a precisão e reduzindo a necessidade de intervenção manual.

4. APLICAÇÕES DA CNN

As CNNs transformaram indústrias com sua alta precisão em tarefas relacionadas à imagem. Elas são amplamente utilizadas em:

- Análise de imagem médica: auxiliando no diagnóstico de doenças através do reconhecimento de imagem.
- Veículos autônomos: permitindo a detecção de objetos e o entendimento da cena.
- Reconhecimento facial: alimentando sistemas de segurança e autenticação.
- Além das aplicações que envolvem imagens, existem pesquisas utilizando CNNs também em processamento de linguagem natural (natural language processing – NLP), interfaces cérebro-computador e séries temporais como financeiras, por exemplo [1].

5. CONCLUSÃO

A arquitetura de uma CNN é projetada para aproveitar a estrutura 2D de uma imagem de entrada. Outro benefício das CNNs é que elas são mais fáceis de treinar e possuem muito menos parâmetros do que redes totalmente conectadas com o mesmo número de unidades ocultas.

As CNNs continuam a ultrapassar os limites do que é possível na aprendizagem profunda, oferecendo soluções poderosas para desafios complexos de dados visuais.

REFERÊNCIAS

[1] CARVALHO JUNIOR, A. Redes Neurais Artificiais: Algoritmos poderosos para aplicações de IA e ML. EAILab Posts. 2024. Disponível em: <https://eailab.labmax.org/2024/04/03/redes-neurais-artificiais-algoritmos-poderosos-para-aplicacoes-de-ia-e-ml/>. Acessado em Ago 13, 2024.

[2] ANALOG DEVICES. What Is Machins Learning? Part 1 – Introduction to convolutional neural networks, 2024. Disponível em:<https://www.radiolocman.com/review/article.html?di=664841>. Acessado em Nov 1, 2024.

[3] SUPERANNOTATE, Convolutional Neural Networks: 1998-2023 Overview, Super Annotate (Blog), 2023. Disponível em: <https://www.superannotate.com/blog/guide-to-convolutional-neural-networks>. Acessado em Ago 13, 2024.

[4] SCIENTISTCAFE, 12.2 Convolutional Neural Network, Scientist Cafe (Blog), 2023. Disponivel em:< https://scientistcafe.com/ids/convolutional-neural-network.html>. Acessado em Ago 13, 2024.

[5] CARVALHO JUNIOR, A. Função de Ativação, o Núcleo da Composição de Neurônios Artificiais. EAILab Posts. 2024. Disponível em: <https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-artificiais/>. Acessado em Ago 13, 2024.

[6] CARVALHO, A., JUSTO, J.F., ANGELICO, B.A. et al., Rotary Inverted Pendulum Identification for Control by Paraconsistent Neural Network, in IEEE Access, 2021. DOI: 10.1109/ACCESS.2021.3080176.

[7] CARVALHO, A., JUSTO, J.F., ANGELICO, B.A. et al. Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum, Applied Soft Computing, 2022, 109927, ISSN 1568-4946. DOI: 10.1016/j.asoc.2022.109927.

[8] LIU, D. A Practical Guide to ReLU. Medium, 2017. Disponível em: <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>. Acessado em Ago 13, 2024.

Elaborado Por: Dr. Arnaldo de Carvalho Junior
Publicado em: Ago 13, 2024
Disponível em: https://eailab.labmax.org/2024/08/13/o-poder-das-cnns-em-aplicacoes-de-ml-envolvendo-identificacao-e-classificacao-de-imagens/. Acessado em Nov 08, 2024.

Post 10: Roteiro Para Criação de Dataset de Imagens Para modelos de Aprendizagem Profunda

Uma das etapas mais importantes para se criar um modelo de aprendizagem profunda (*deep learning* – DL) é criar um *dataset* (conjunto de dados) de imagens em grande escala para treinar o modelo.

Mas como coletar, rotular e armazenar imagens de maneira eficiente e eficaz?

Neste post será abordado o passo a passo do processo de criação do *dataset* de imagem, incluindo desde o planejamento e preparação até os detalhes de anotação e organização, conforme a Figura 1.

Figura 1 – Fluxograma de Criação de Dataset de Imagens.
Fonte: Autoria Própria (2024).

1. COMPREENDENDO AS NECESSIDADES DO *DATASET* DE IMAGEM

Quando se trata de construir um *dataset* de imagem, o primeiro passo é entender o propósito que o *dataset* servirá. Se o propósito é para pesquisa, um modelo de inteligência artificial (IA) ou DL, ou diversão, saber o propósito ajudará a determinar que tipo de dados são necessários e quanto deles.

Por exemplo, se o objetivo do *dataset* de imagem for um projeto de pesquisa, os dados devem ser de alta qualidade e ter um nível específico de detalhe que auxiliará no estudo.

Por outro lado, se o propósito é apenas para diversão, então o *dataset* pode ser mais relaxado, variado e aberto. Se um modelo de IA ou DL usar o *dataset*, ele deverá ser **estruturado** para facilitar o aprendizado e o **reconhecimento de padrões** nas imagens pelo sistema. Portanto, a clareza sobre o propósito das imagens é crucial na construção de um *dataset* bem-sucedido e eficaz.

Ao construir um dataset de imagem, uma das primeiras coisas a considerar é o tamanho e a complexidade dos dados. Como tal, é importante determinar quantas imagens são necessárias e quais tipos de imagens atendem às necessidades.

É importante lembrar que o tipo de imagens no *datase*t impactará significativamente o quão **preciso** e **eficaz** será qualquer modelo de aprendizado de máquina (*machine learning* – ML) que se treinar nele.

Portanto, gastar tempo pesquisando e planejando o tamanho e a complexidade ideais do *dataset* de imagens é crucial para obter resultados confiáveis.

Em seguida, ao fazer um *dataset* de imagem, é preciso considerar os requisitos de qualidade que as imagens precisam atender para fornecer resultados precisos.

A resolução das imagens é uma grande parte se descobrir o quão bom é o *dataset*. Podem ser necessárias imagens de alta ou baixa resolução dependendo do foco da pesquisa ou projeto. Além disso, incluir recursos específicos, como cor, orientação e outras características visuais, é fundamental.

Esses recursos ajudam a criar um *dataset* mais abrangente e detalhado que pode ser utilizado para diversos fins. O desenvolvimento de um *dataset* de imagem requer uma consideração cuidadosa de seus requisitos e recursos de qualidade, o que garante que ele atenda ao propósito pretendido.

2. PREPARANDO RECURSOS PARA A EMPREITADA

Como a importância do *dataset* de imagens continua a crescer em diferentes áreas de pesquisa, é essencial preparar recursos para dimensionar esses conjuntos de dados. Isso

envolve entender as várias ferramentas e técnicas disponíveis para gerenciar e armazenar grandes quantidades de dados.

a) Gerenciamento: Os conjuntos de dados de imagens requerem gerenciamento adequado para garantir seu uso prático, seja para visão computacional, ML ou outras aplicações.

b) Preparação: Gerenciar e preparar dados de imagem requer uma consideração cuidadosa de fatores como **anotação**, **rotulagem** e **limpeza de dados**.

c) Armazenamento: Um aspecto crucial do processo de gerenciamento é selecionar os métodos de armazenamento mais adequados, como sistemas de arquivos distribuídos ou armazenamento em nuvem, que possam lidar com eficiência com o volume e a complexidade dos dados. O armazenamento em nuvem não apenas torna os dados mais fáceis de encontrar e acessar, mas também torna os dados mais seguros e confiáveis.

d) Segurança: Com o armazenamento em nuvem, os dados são criptografados e protegidos contra acesso não autorizado. Isso reduz o risco de violações de dados.

e) Ferramentas de Otimização: A escala de *dataset* de imagens requer uma compreensão abrangente das ferramentas e técnicas necessárias para otimizar o armazenamento, processamento e preparação de dados para análise.
As plataformas de armazenamento em nuvem oferecem soluções que podem ser ampliadas à medida que o armazenamento de dados cresce. Esforços colaborativos entre pesquisadores e partes interessadas também são possíveis, pois o armazenamento em nuvem permite fácil compartilhamento e colaboração em um ambiente seguro. À medida que o volume de conjuntos de dados de imagem aumenta, aproveitar o armazenamento em nuvem continua sendo crucial para gerenciar e armazenar arquivos grandes, mantendo a segurança e a acessibilidade dos dados.

3. ESTRATÉGIAS DE AQUISIÇÃO DE IMAGENS

A aquisição de imagens de alta qualidade é um passo essencial na construção de um *dataset* de imagens. Requer consideração cuidadosa de vários fatores, como fonte, formato e tamanho das imagens.

a) Fonte: A origem das imagens deve ser confiável para garantir autenticidade e credibilidade. Além disso, o formato deve ser compatível com o *dataset* e as ferramentas que serão utilizadas para análise.

Nota: Em post anterior (CARVALHO, 2024), foram publicados referências de *dataset* disponibilizados publicamente para diversas áreas do conhecimento humando. Podem ser um bom ponto de partida de construção de *dataset*.

b) Tamanho das Imagens: é outro fator crítico que deve ser considerado. Imagens grandes podem consumir recursos significativos de memória, enquanto imagens pequenas podem precisar fornecer mais detalhes para análise. Portanto, é necessário equilibrar a qualidade da imagem e o tamanho do arquivo. A seleção de imagens deve ser abrangente e representativa para garantir que o *dataset* represente com precisão a população ou os fenômenos pretendidos.

c) Aquisição de imagens: é um processo crítico que requer deliberação cuidadosa e atenção aos detalhes para garantir que o *dataset* seja preciso e confiável. O processo de aquisição deve incluir de forma abrangente as variações de ângulos, condições de iluminação e objetos.

Por fim, criar um *dataset* de imagens é crucial em muitos campos, como visão computacional e ML. Isso pode ser conseguido através da aquisição manual ou automatizada de imagens de diversas fontes, como bancos de dados *on-line*, repositórios, pesquisas estruturadas na web ou configurações personalizadas de câmeras.
É importante notar que a criação de um *dataset* de imagem não consiste apenas em coletar, mas também em garantir que eles sejam de alta qualidade, diversificados e relevantes para a tarefa. Depois que o *dataset* de imagem é criado, ele pode ser usado para diversas aplicações, como reconhecimento de objetos, classificação de imagens e análise de cena. Concluindo, a criação de um *dataset* é fundamental para o avanço do campo da visão computacional e do ML.

4. PRÉ-PROCESSAMENTO E PREPARAÇÃO DE IMAGENS

Ao se trabalhar em um projeto com um conjunto de dados de imagens, é essencial configurar uma maneira confiável de lidar com os dados.

O pré-processamento e a preparação são etapas integrais para atingir esse objetivo. Essas etapas envolvem a **manipulação** e **otimização** das imagens para garantir que eles estejam em um formato que seja propício aos requisitos do projeto.

Durante a fase de pré-processamento, tarefas como **filtragem, redimensionamento e normalização** geralmente são feitas para garantir que todas tenham a mesma qualidade e formato. Esta etapa garante que os dados estejam prontos para análise, permitindo que os pesquisadores extraiam ***insights*** dos dados de forma mais eficaz.

O pré-processamento faz também com que os **modelos de ML** mais precisos, o que os torna melhores em tarefas como reconhecimento. Um bom pré-processamento é uma das etapas mais críticas para garantir que um projeto de conjunto de dados atenda aos seus objetivos e atenda a altos padrões de qualidade e precisão.

Além disso, é imperativo observar que o manuseio adequado de um *dataset* é crucial para seu uso bem-sucedido em aplicativos de ML e visão computacional.

O **redimensionamento** e **recorte** durante a fase de preparação devem ser realizados com muito cuidado para garantir que os dados representem com precisão os objetos e cenas dos quais foram inicialmente retirados.

É importante ao usar o *dataset* para tarefas como detecção ou classificação de objetos, observar que a integridade do sistema afeta diretamente a precisão dos algoritmos. Para fazer um *dataset* confiável e útil, deve-se prestar muita atenção a cada detalhe durante a preparação.

5. VALIDAÇÃO E GARANTIA DA QUALIDADE

O *dataset* é crucial para muitos setores e campos de pesquisa, incluindo ML, visão computacional e IA.

a) Verificação: Para garantir dados da mais alta qualidade para processos de validação e garantia de qualidade, é crucial examinar minuciosamente e verificar a precisão de cada *dataset* antes de usar. Nesse processo, cada um é analisado, seus metadados são verificados e garante que atenda a padrões específicos. O *dataset* deve ser preciso e consistente para que os modelos de ML sejam confiáveis e robustos.

b) Testes de Qualidade e Validação: Dados de má qualidade podem levar a previsões imprecisas, o que pode ser prejudicial ao desempenho geral do sistema. Portanto, é essencial priorizar a qualidade dos dados ao trabalhar com conjuntos de dados para

alcançar resultados precisos e confiáveis. Ao lidar com *dataset*, é crucial garantir que os dados dentro deles atendam aos padrões de qualidade esperados. Uma maneira de garantir isso é executando testes automatizados no conjunto de dados.

c) Identificação de Anomalias e Inconsistências: Isso ajuda a identificar quaisquer anomalias ou inconsistências nos dados, que podem então ser abordadas antes que se tornem problemas significativos. Recomenda-se também a realização de testes manuais para verificar a precisão de quaisquer detalhes adicionais incluídos no conjunto de dados, como rótulos ou anotações. Isso é especialmente importante para aplicações sensíveis ou essenciais, como imagens médicas ou carros autônomos. Aproveitar o tempo para testar e confirmar a precisão de um *dataset* pode economizar tempo e dinheiro no longo prazo e torná-lo mais confiável e confiável em geral.

d) Manutenção Periódica: É essencial verificar o conjunto de dados com frequência para garantir que sua precisão permaneça a mesma ao longo do tempo. Mudanças podem acontecer nos conjuntos de dados por vários motivos, como quando os dados são corrompidos ou surgem novas tecnologias. Testes de validação e garantia de qualidade devem ser feitos regularmente e quaisquer alterações necessárias devem ser feitas para manter resultados precisos. Além disso, recomenda-se estabelecer um protocolo para manutenção e atualização regular de *dataset.* Essa prática garante que os usuários do *dataset* tenham acesso a dados confiáveis e atualizados.

Seguindo um processo de manutenção bem planejado, as organizações podem usar seus *datasets* como uma ferramenta confiável para obter *insights* e tomar decisões inteligentes. Em última análise, o sucesso de qualquer *dataset* depende do compromisso com a sua melhoria e manutenção contínuas.

6. PÓS-PROCESSAMENTO E FINALIZAÇÃO DO *DATASET*

Um *dataset* é uma coleção usada para diversos fins, incluindo treinamento de algoritmos de ML ou realização de pesquisas.

O pós-processamento do *dataset* é a etapa final e crucial na preparação do conjunto de dados para uso. Envolve um conjunto de etapas de garantia de qualidade que garantem a **correção**, **precisão** e **confiabilidade** dos dados.

O pós-processamento inclui várias técnicas, como **correção**, **normalização de dados**, **filtragem** e **segmentação**. Esses métodos garantem que sejam de alta qualidade,

livres de ruído e mantenham o nível de consistência desejado. Por causa disso, o pós-processamento desempenha um papel significativo para garantir que o *dataset* esteja pronto para ser usado, confiável e útil no trabalho em direção à meta estabelecida para ele. As tarefas de pós-processamento, como **rotulagem**, **categorização** e **criação de metadados** podem ser tediosas e demoradas, mas são fundamentais para tornar o *dataset* valioso para aplicativos de ML e visão computacional.

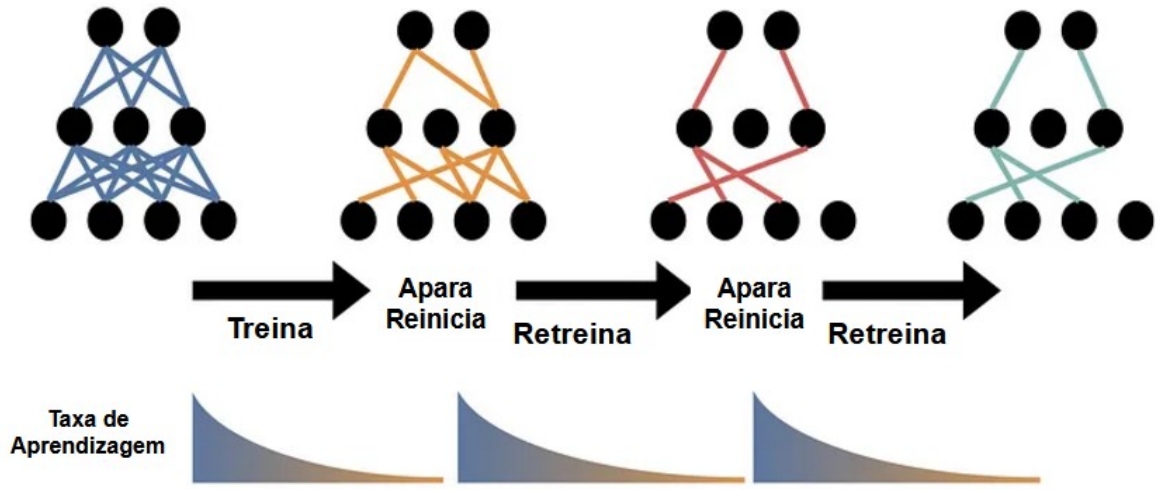


Figura 2: Validação do *Dataset* para IA.
Fonte: Adaptado de XOFFSHORINGT (2024).

O *dataset* é melhorado tanto em termos da sua qualidade como da sua confiabilidade após a eliminação do duplicado e a execução de verificações de qualidade. Os metadados fornecem informações essenciais sobre e contribuem para resultados de pesquisa mais precisos e úteis. O pós-processamento é uma etapa essencial na criação de um conjunto de dados preciso e valioso que pode ser usado em vários contextos e aplicações.

7. CONCLUSÃO

Criar um *dataset* em grande escala para os modelos de DL pode inicialmente parecer assustador. Ainda assim, se for empregada a estratégia apropriada, pode-se concluir a tarefa de forma rápida e eficiente. Se este guia for seguido, o resultado será um *dataset* que contribuirá para o desenvolvimento de modelos de ML confiáveis com mais frequência. Além disso, a ordem e a limpeza em relação a tudo e qualquer coisa associada à anotação e armazenamento estarão mantidos. Seguindo as diretrizes descritas neste guia pode-se construir um *dataset* que impulsionará significativamente os esforços de DL.

REFERÊNCIAS

XOFFSHORINGT, A Comprehensive Guide to Creating a Large-Scale Image Dataset for Deep Learning Models, Medium, 2024. Disponível em: <https://medium.com/@24x7offshoringt/a-comprehensive-guide-to-creating-a-large-scale-image-dataset-for-deep-learning-models-e2922a9f36b1>. Acessado em Ago 09, 2024.

24x7OFFSHORING. A Comprehensive Guide to Creating a Large-Scale Image Dataset for

Deep Learning Models? 2021. Disponível em: <https://24x7offshoring.com/a-comprehensive-guide-to-creating-a-large-image/?feed_id=32040&_unique_id=65dff239aa281>. Acessado em Ago 09, 2024.

CARVALHO, A. Datasets de Acesso Livre para Projetos de IA!, Posts, EAILAB, IFSP, Jun 17, 2024. Disponível em: https://eailab.labmax.org/2024/06/17/datasets-de-acesso-livre-para-projetos-de-i-a/. Acessado em Set 24, 2024.

Elaborado por: Dr. Arnaldo de Carvalho Junior

Publicado em: Set 24, 2024

Disponível em: https://eailab.labmax.org/2024/09/24/roteiro-para-criacao-de-dataset-de-imagens-para-modelos-de-aprendizagem-profunda/. Acessado em Nov 08, 2024.

ANEXO 1 - Pipeline de Classificação de Imagens com Inteligência Artificial

Projeto de iniciação científica (IC) desenvolvido pelo bacharelando em engenharia **Bruno Gobato Simões**, do IFSP campus Cubatão, intitulado ***"Inteligência Artificial em computação de borda: Elaboração de um pipeline para classificação de imagens"***, em 2024, sob orientação do **Dr. Arnaldo de Carvalho Junior** e **Dr. Walter Augusto Varella** resultou na elaboração e publicação do **Pipeline de Classificação de Imagens com Inteligência Artificial** , na plataforma Github. A Figura 1 apresenta a tela principal do Pipeline.

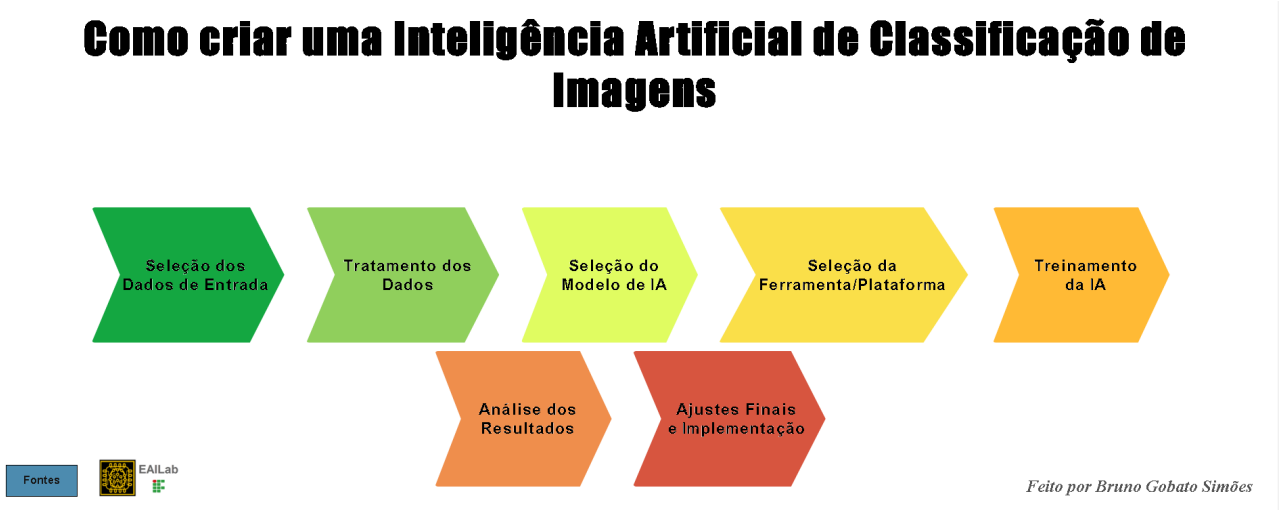


Figura 1 – Página Principal do Pipeline de Classificação de Imagens com IA.
Fonte: O Autor (2024).

O Pipeline é totalmente interativo, podendo ser acessado através do QR-Code apresentado na a seguir, ou pelo link: https://eailab-ifsp.github.io/AI-CLASSIFICATION-PIPELINE/