Uma exploração de redes neurais biologicamente realistas

Filipe Rodrigues Batista de Oliveira
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
filipe.oliveira@dcc.ufmg.br

Antônio Carlos Guimarães de Almeida
Departamento de Engenharia de Biossistemas
Universidade Federal de São João Del Rey
São João Del Rey, Brasil
acga@ufsj.edu.br

Omar Paranaiba Vilela Neto
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
omar@dcc.ufmg.br

***Abstract*—Esse trabalho apresenta uma introdução ao problema do "esquecimento catastrófico", suas consequências e como a utilização de redes neurais artificiais que incorporem mecanismos observados no cérebro podem ajudar a resolver esse problema e melhorar a eficiência no treinamento destes modelos.**

***Index Terms*—neural networks, computational neuroscience, machine learning**

I. Introdução

Um dos grandes problemas enfrentados atualmente no treinamento de modelos de aprendizado de máquina, consiste no chamado "esquecimento catastrófico" que é um fenômeno em inteligência artificial, inicialmente observado por McCloskey e Cohen em 1989 no qual redes neurais artificiais tendem a esquecer informações aprendidas anteriormente ao serem treinadas para novas tarefas. Isso ocorre porque o processo de aprendizado de novas informações pode sobrescrever ou interferir nos pesos de conxeão entre os neurônios em cada camada de uma rede já previamente treinada, resultando em um desempenho significativamente pior em tarefas já aprendidas [5] atrapalhando o avanço tecnológico em áreas onde os modelos são constantemente atualizados, como na Robótica, por exemplo.

Além disso, considere que há um grande gasto energético despendido tanto para o treinamento dos modelos quanto para o resfriamento das máquinas utilizadas para tal atividade, logo re-treinar os modelos para cada nova tarefa configuraria alto grau de desperdício de energia. E em um cenário de mudanças climáticas que são provocadas principalmente pelo processo de produção de energia, isso se torna inaceitável.

Em relação aos modelos utilizados atualmente de redes neurais artificiais, os detalhes biofísicos (mecanismos não sinápticos) dos neurônios reais não são levados em consideração nos modelos computacionais, e isso ocorre pois há um aumento na complexidade computacional que torna o treinamento desses modelos mais custosos de serem utilizados na prática, apesar disso, como referenciado por [3] existem trabalhos que sugerem que essas características atualmente negligênciadas, tem papel importante no processamento da informação e consequentemente no aprendizado, sendo uma possível saída para melhorar os sistemas inteligentes atuais e talvez solucionar o problema apontado no primeiro parágrafo.

Usamos o livro [1] como referência para o aprendizado de alguns conceitos de Neurociência e [3] como trabalho norteador e, após essa seção iremos fazer uma breve introdução aos mecanismos neurais. Em seguida, vamos apresentar o que foi feito.

II. Referencial teórico

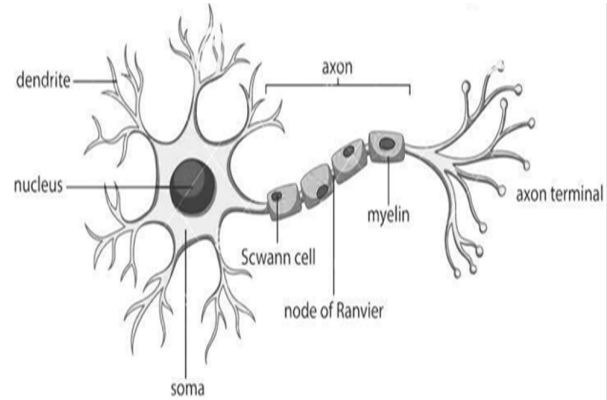

Fig. 1. Diagrama esquemático de um neurônio (Fonte: [1])

Em redes neurais reais a informação é propagada graças a processos biofísicos de mudança na concentração de íons (elementos químicos eletricamente carregados) dentro do corpo do neurônio em relação ao meio extracelular. Inicialmente, íons de potásio (K+) estão mais concentrados dentro da célula, enquanto temos uma maior concentração de íons de Sódio (Na+) e cloreto (Cl-) no meio extracelular,e a membrana

plasmática do neurônio (estrutura que separa o meio intracelular do extracelular) é dito estar no seu potencial de repouso.

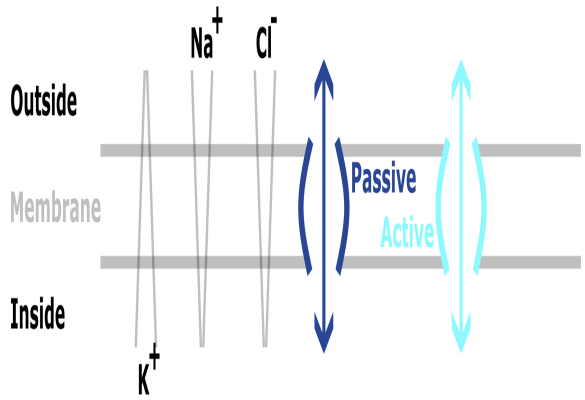

Fig. 2. Diagrama mostrando movimentos iônicos na membrana do neurônio (Fonte: [1])

Quando certos neurotransmissores (moléculas químicas) liberados por outros neurônios se ligam a receptores presentes nos dendritos (veja a figura 1) de um neurônio, canais (portas presentes na membrana plasmática de uma célula que permitem a passagem de tipos específicos de elementos) de Na+ se abrem na membrana permitindo a passagem de íons de Na+ para o meio intracelular e o respectivo aumento da concentração desta espécie química no mesmo; Quando a concentração deste tipo de íon atinge um certo limiar, canais sensíveis a voltagem de Na+ se abrem, permitindo que mais Na+ entre na célula, o que faz com que o potencial elétrico da membrana chegue a um valor máximo.

Ao mesmo tempo, canais de K+ se abrem, fazendo com que potásio saia da célula, reduzindo o potencial da membrana, além de que, em altas voltagens canais de Na+ se fecham. Como o tempo para o fechamento dos canais de K+ é maior do que os de Na+, o potencial de membrana diminui para um valor abaixo do seu potencial de equilibrio, esse período é chamado de período refratário, pois efetivamente neste momento é mais difícil para a célula emitir um potencial de ação (uma resposta).

Em seguida, uma combinação de movimentos iônicos passivos e ativos (bomba de sódio e potássio) restauram o valor do potencial de membrana para o seu potencial de equilibrio.

Os interneurônios são a mais numerosa classe de neurônios, sendo encontrados somente no sistema nervoso central, participando no processo de processamento da informação, atuando como um elo entre outros tipos de neurônio. Eles são subdivididos por sua vez em, locais e de retransmissão, os locais tem axônios curtos e formam circuitos com neurônios próximos para analisar pedaços de informação, enquanto os de retransmissão, conectam circuitos de neurônios de uma região do cérebro para outra. [6] Em nosso trabalho, usamos como inspiração o primeiro tipo.

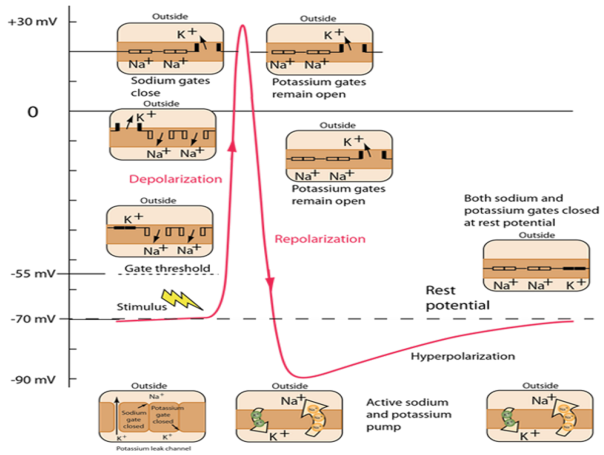

Fig. 3. Diagrama mostrando o valor do potencial de membrana e as respectivas fases: polarização, depolarização e hiperpolarização (Fonte: [1])

III. Metodologia

Usamos como base para o desenvolvimento e também usaremos para comparação com a implementação final, um Multilayer Perceptron. Com objetivo de facilitar o trabalho e de economizar tempo, optou-se por procurar uma implementação open-source [4] deste modelo que fosse composto por poucos neurônios, que tivesse poucas camadas e que o implementasse do ”zero”, de forma a possibilitar a modificação do código já pronto afim de acrescentar algumas das pecularidades e dinâmicas observadas em redes neurais biológicas.

O problema que resolvemos (com o intuito de observar o comportamento da rede modificada em comparação com a abordagem tradicional) é o clássico problema de classificação de digitos manuscritos; Para isso, usamos a base de dados MNIST do framework Keras, composto de 1797 imagens monocromáticas de digitos de 0 a 9, sendo que cada imagem tem 8x8 pixels.

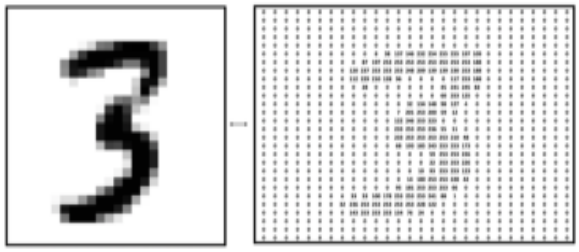

Fig. 4. Uma instância do MNIST (algarismo 3) e sua respectiva representação matricial

Inicialmente foi proposto uma arquitetura de três camadas, aonde o interneurônio estaria conectado com todos os neurônios de todas as camadas (exceto a camada de entrada). Porém, se feito dessa forma, nós obteríamos uma RNN (Recurrent Neural Network) e não uma feedforward neural network (tipos de redes neurais cuja informação move-se em apenas

uma direção, em suma, não existem loops), o que não era a intenção, pois dessa forma, não poderíamos usar o algoritmo de Backpropagation, já que quebrariamos uma das premissas que o mesmo assume (de que a rede seja feedforward). Para evitar esse problema então, o interneurônio só seria excitado pelo neurônios da camada de entrada, enquanto inibiria todos os outros.

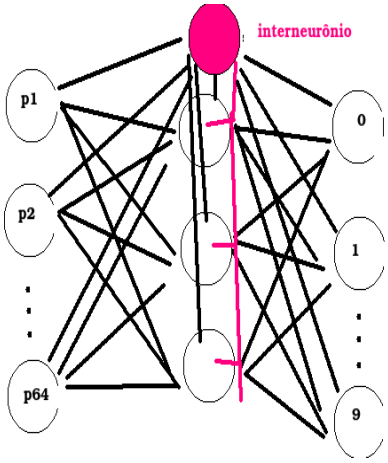

Fig. 5. Arquitetura proposta inicialmente, de vermelho temos o interneurônio e as suas respectivas conexões inibitórias.

Como é possível ver na figura 5, nós acrescentamos mais uma camada a rede, pois a presença de apenas três camadas não teria poder de representação suficiente para resolver o problema de detecção de dígitos.

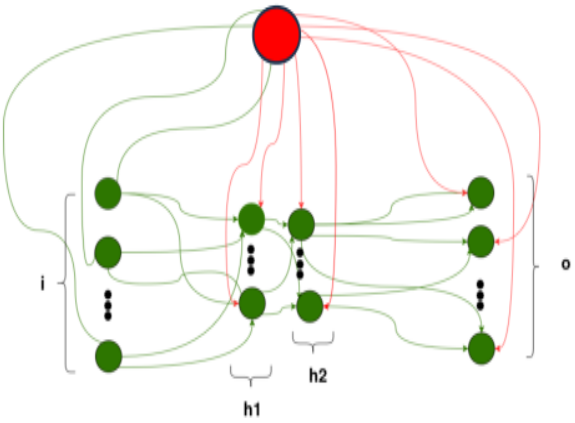

Fig. 6. Arquitetura implementada, de verde temos os neurônios e as respectivas conexões excitatórias e em vermelho temos o interneurônio e as suas respectivas conexões inibitórias.

O que distingue a nossa solução de uma MLP convencional (veja a figura 5) é:

1) A inclusão do interneurônio, isto é, um neurônio (no contexto de aprendizado de máquina) que apenas pode inibir (conexão com valor negativo) os outros neurônios, mas que por sua vez é excitado (conexão com valor positivo) por outros neurônios.
2) Em toda a rede, exceto do interneurônio para os neurônios, todas as conexões devem ser não negativas.

A razão pela qual há apenas um interneurônio é que no hipocampo cerebral a quantidade dos mesmos estão expressos na razão de três interneurônios para cada cem neurônios.

Para (1), foram implementadas duas versões: a primeira considera que o mesmo atua continuamente, enquanto a segunda, mais fidedigna do ponto de vista biológico, só dispara um potencial de ação, isso é, terá o seu valor acrescentado na etapa de propagação, quando atingido um certo valor. Esse valor é um hiperparâmetro da rede e deve ser ajustado manualmente. Em ambas as versões, é possível definir taxas de aprendizado diferentes para os pesos do interneurônio e dos neurônios, bem como para os pesos inibitórios e excitatórios do interneurônio.

Para manter o invariante (2) nós verificamos se há pesos de valores negativos e, caso a resposta seja afirmativa, nós simplesmente multiplicamos os mesmos por -1.

Usamos como função de ativação a ReLU: $a(z) = max(0, z)$, aonde $z \in \mathbb{R}^n$, para neurônios e para o interneurônio, e a softmax para os neurônios da camada de saída: $\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$, aonde $z \in \mathbb{R}^k$ já que o problema em questão é um problema de classificação com mais de duas classes.

Implementamos a versão do stocastic gradient descent com mini-batchs de tamanho 32, seguindo boas práticas da indústria de se usar potências de 2 para isso e adaptamos o algoritmo de backpropagtion para o cálculos dos gradientes também em relação aos pesos inibitórios.

Como função de perda nós usamos o MSE (Mean squared error): $\epsilon = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, onde y é o valor real e $\hat{y}$ o valor previsto pela rede e n a quantidade de dados.

Nós capturamos como métricas a loss e a acurácia para os conjuntos de teste e treinamento.

Foram realizamos três experimentos, que serão apresentados a seguir.

IV. Experimentos

Foi definido como alvo uma acurácia de 95% nos dados de treino.

Nós variamos os valores das taxas de aprendizado para os pesos excitatório e inibitórios do interneurônio e excitatório do neurônio para diferentes valores, bem como para valores de threshold para a emissão dos potenciais de ação dos mesmos. Também alteramos a quantidade de neurônios das camadas escondidas afim de verificar o seu efeito no aprendizado.

Os melhores valores encontrados estão apresentados nos gráficos a seguir, que apresentam na parte superior os dados referentes ao conjunto de treino e abaixo os resultados referentes ao conjunto de teste.

Em nossos experimentos nós iteramos sobre os dados de entrada usando 100 epochs. Perceba que os modelos implementados sempre aprenderam (ligeiramente) mais rápido que a MLP convencional, embora nenhum deles tenha atingido o alvo definido. Além disso, outro fato interessante a se notar e passível de futuras investigações, é que não houve melhoria entre inibir continuamente os outros neurônios em comparação

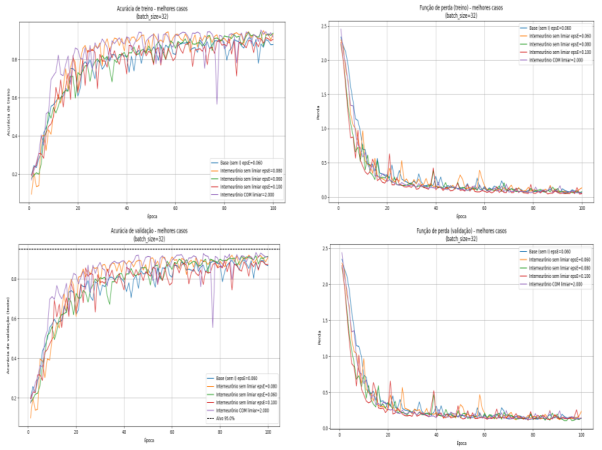

Fig. 7. Métricas coletadas para experimentos, aonde as camadas ocultas possuem 5 neurônios.

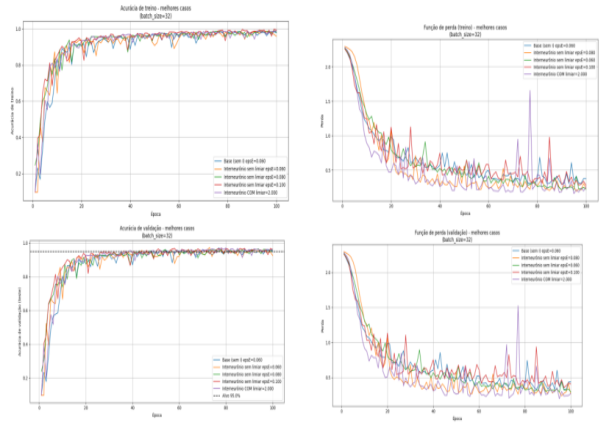

Fig. 8. Métricas coletadas para experimentos, aonde as camadas ocultas possuem 64 neurônios.

com ao mecanismo de disparo: ambos apresentaram resultados similares.

V. Conclusão

Um dos maiores desafios encontrados foi adaptar o algoritmo de Backpropagation para acomodar a modificação arquitetural feita (inclusão do interneurônio).

E embora o acréscimo do interneurônio tornou a rede mais rápida para o aprendizado (como podemos ver na seção que precede essa em todos os casos), devido a simplicidade da base de dados escolhida não foi possível perceber uma diferença tão significativa com relação a MLP, sendo necessário testes com outros problemas mais complicados. Além disso, seria necessário avaliar também se o modelo é mais robusto ao fenômeno de ”esquecimento catastrófico”.

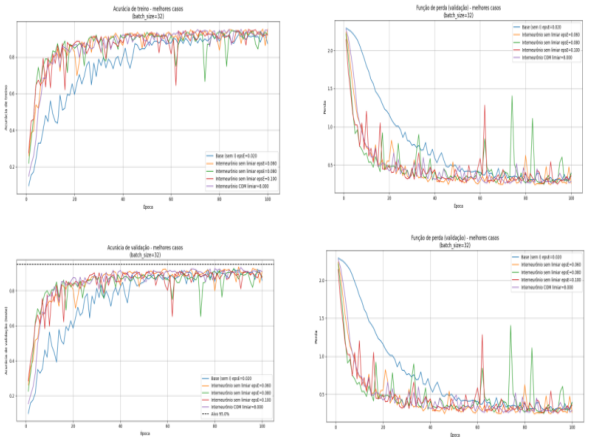

Fig. 9. Métricas coletadas para experimentos, aonde a primeira camada oculta possui 5 neurônios e a segunda possui 64 neurônios.

References

[1] Goodman, Dan F. M., and Marcus Ghosh. ’Neuroscience for Machine Learners’. 12 December 2023. https://doi.org/10.5281/zenodo.10366802.G.

[2] A. Shrestha, H. Fang, Z. Mei, D. P. Rider, Q. Wu and Q. Qiu, ”A Survey on Neuromorphic Computing: Models and Hardware,” in IEEE Circuits and Systems Magazine, vol. 22, no. 2, pp. 6-35, Secondquarter 2022, doi: 10.1109/MCAS.2022.3166331. keywords: Neuromorphic engineering;Computational model- ing;Biological system modeling;Neurons;Computer architec- ture;Hardware;Encoding

[3] Depannemaecker D, Canton Santos LE, Rodrigues AM, Scorza CA, Scorza FA, Almeida AG. Realistic spiking neural network: Non-synaptic mechanisms improve convergence in cell assembly. Neural Netw. 2020 Feb;122:420-433. doi: 10.1016/j.neunet.2019.09.038. Epub 2019 Oct 16. PMID: 31841876.

[4] Implementing a Neural Network from Scratch in Python. Disponível em: https://github.com/dennybritz/nn-from-scratch

[5] Aleixo, E. L., Colonna, J. G., Cristo, M., & Fernandes, E. (2024). Catastrophic Forgetting in Deep Learning: A Comprehensive Taxonomy. Journal of the Brazilian Computer Society, 30(1), 175–211. https://doi.org/10.5753/jbcs.2024.3966

[6] Kandel, Eric; Schwartz, James; Jessell, Thomas, eds. (2000). Principles of Neural Science 4th ed. New York City, New York: McGraw Hill Companies. p. 25. ISBN 978-0-8385-7701-1