UNIVERSIDADE DO ESTADO DE MATO GROSSO
CÂMPUS UNIVERSITÁRIO DE SINOP
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL – PROFMAT

DOUGLAS SALES PAULI

A MATEMÁTICA NAS REDES NEURAIS

Sinop/MT

2024

DOUGLAS SALES PAULI

A MATEMÁTICA NAS REDES NEURAIS

Dissertação apresentado à Banca Examinadora do Mestrado Profissional em Matemática em Rede Nacional – PROFMAT da Universidade do Estado de Mato Grosso, Câmpus Universitário de Sinop, como pré-requisito para a obtenção do título de Mestre em Matemática.

Prof. Dr. Raul Abreu de Assis
Orientador

Profa. Dra. Luciana Mafalda Elias de Assis
Coorientadora

Sinop/MT

2024

A Deus, pela vida. Minha esposa e família, pelo apoio e compreensão. Aos meus professores e amigos de pós-graduação.

Ficha catalográfica elaborada pela Supervisão de Bibliotecas da UNEMATCatalogação de Publicação na Fonte.
UNEMAT - Unidade padrão

Pauli, Douglas Sales.
A MATEMÁTICA NAS REDES NEURAIS / Douglas Sales Pauli. - Sinop, 2024.
77f.: il.

Universidade do Estado de Mato Grosso "Carlos Alberto Reyes Maldonado", Matemática/SNP-PROFMAT - Sinop - Mestrado Profissional, Campus Universitário De Sinop.
Orientador: Raul Abreu de Assis.
Coorientador: Luciana Mafalda Elias de Assis.

1. Redes Neurais. 2. inteligência artificial. 3. palestras como produto educacional. I. Assis, Raul Abreu de. II. Assis, Luciana Mafalda Elias de. III. Título.

UNEMAT / MTCDU CDU 004

DOUGLAS SALES PAULI

A MATEMÁTICA NAS REDES NEURAIS

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática em Rede Nacional – ProfMat da Universidade do Estado de Mato Grosso/UNEMAT – Campus Universitário de Sinop, como requisito parcial para obtenção do título de Mestre em Matemática.

Orientador(a): Prof. Dr. Raul Abreu de Assis
Coorientador(a): Profa. Dra. Luciana Mafalda Elias de Assis
Aprovado em 02/08/2024

BANCA EXAMINADORA

Prof. Dr. Raul Abreu de Assis
UNEMAT – SINOP - MT

Profa. Dra. Chiara Maria Seidel Luciano Dias
UNEMAT – SINOP - MT

Prof. Dr. Rubens Pazim Carnevarollo Júnior
UFMT - SINOP - MT

Sinop/MT
2024

AGRADECIMENTOS

Agradeço a Deus por ter me dado a vida, saúde e total capacidade de chegar até aqui.

A minha esposa pelo apoio, e pelo ombro que me apoiou nos momentos que mais precisei, sou privilegiado por morar com alguém que estuda as mesmas coisas que eu e tem os mesmos sonhos profissionais. A experiência de ter com quem conversar sobre o que tenho aprendido, muitas vezes tem sido meu norte e que também está passando por este processo de estudo ao meu lado.

Aos meus pais por terem me apoiado, muito antes de eu ter ingressado neste curso de pós-graduação, sempre me auxiliando em todas as minhas decisões que me trouxeram até este dia. Pela capacidade de acreditar em mim e terem sonhado esta realização ao meu lado.

Agradeço imensamente aos meus colegas de pós-graduação, companheiros incansáveis nesta jornada acadêmica. Juntos, enfrentamos desafios e compartilhamos conquistas. Valorizo cada debate enriquecedor sobre educação e das várias horas de estudo que passamos juntos. E um brinde a disciplina optativa obrigatória que não permitiu que enlouquecêssemos.

Expresso minha profunda gratidão ao meu orientador, professor Dr. Raul Abreu de Assis, pelo inestimável apoio e orientação. Sua confiança e paciência foram fundamentais durante o processo de elaboração desta dissertação, e sua sabedoria guiou-me por caminhos que levaram à excelência acadêmica e ao sucesso deste trabalho.

Minha sincera gratidão é dirigida aos estimados professores que foram pilares fundamentais da minha jornada no mestrado profissional. Cada aula ministrada foi um degrau a mais em minha escalada acadêmica, e as contribuições de vocês foram essenciais para moldar o profissional que me tornei. Obrigado por compartilharem seu conhecimento e por inspirarem não só a minha carreira, mas também a minha visão de mundo.

Muito obrigado a todos.

“Comece fazendo o que é necessário, depois o que é possível, e de repente você estará fazendo o impossível.” São Francisco de Assis

RESUMO

Neste trabalho é apresentado uma proposta de estudo e uma análise detalhada da matemática subjacente aos algoritmos de redes neurais aplicados ao reconhecimento de imagens. Explora-se a complexidade e a elegância das equações e funções que permitem que essas redes 'aprendam' a identificar padrões visuais. Adicionalmente, o trabalho sugere uma iniciativa de divulgação científica para alunos do ensino médio. Através de palestras na cidade de Lucas do Rio Verde, Mato Grosso, buscou-se despertar o interesse dos jovens pela interseção entre inteligência artificial e matemática, demonstrando como conceitos abstratos se traduzem em tecnologias inovadoras com aplicações práticas.

Palavras chave: Redes Neurais, inteligência artificial, palestras como produto educacional.

ABSTRACT

This work presents a study and a detailed analysis of the mathematics underlying neural network algorithms applied to image recognition. It explores the complexity and elegance of the equations and functions that enable these networks to 'learn' to identify visual patterns. Additionally, the paper suggests a scientific divulgation initiative for high school students. Through lectures in the city of Lucas do Rio Verde, Mato Grosso, we sought to awaken the interest of young people in the intersection between artificial intelligence and mathematics, demonstrating how abstract concepts translate into innovative technologies with practical applications.

Keywords: Neural Networks, artificial Intelligence and Lectures as an educational product

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

O debate sobre redes neurais não é recente, sendo um tópico de pesquisa muito ativo pelo menos nos últimos 40 anos. Para alguns, essas redes parecem saídas de obras de ficção científica, mas na realidade, elas representam uma tecnologia emergente que, após um período de desenvolvimento latente, encontrou aplicações práticas significativas. A jornada começou nos anos 1940 com a busca por um modelo computacional que emulasse o funcionamento das células cerebrais, impulsionada pela pesquisa pioneira de Mcculloch e Pitts (1943). Após um declínio nos estudos durante os anos 1970, a década de 1980 testemunhou um renascimento do interesse, graças a avanços metodológicos e ao aumento da capacidade computacional.

Redes neurais são um ramo da ciência da computação intimamente ligado à inteligência artificial, com o objetivo de desenvolver modelos matemáticos que imitem as estruturas neurais biológicas. O desafio é programar computadores para replicar comportamentos inteligentes, como jogar xadrez, manter diálogos, ou resolver problemas complexos. A expectativa é que, ao modelar algoritmos com base nos circuitos cerebrais, comportamentos inteligentes surjam, aprendendo e evoluindo através de erros e descobertas.

A capacidade humana de executar tarefas complexas e aprender novas habilidades deriva do processamento paralelo e distribuído da rede de neurônios no cérebro. Especificamente, os neurônios do córtex, responsáveis pelo processamento cognitivo, adaptam-se a novos conhecimentos e experiências por meio de mudanças estruturais, reconfigurando as redes neuronais e modificando sinapses (HAYKIN, 2007, p.32-36).

As redes neurais artificiais são sistemas inspirados na arquitetura cerebral humana, capazes de aprender autonomamente a partir de exemplos e executar tarefas complexas. Compostas por uma vasta rede de neurônios artificiais interligados, elas processam informações e tomam decisões baseadas em padrões e dados. Essas redes encontram aplicação em uma variedade de campos, incluindo reconhecimento de voz e imagem, previsão de séries temporais e classificação de dados.

Embora as redes neurais tenham sido inicialmente confinadas ao âmbito acadêmico, a ascensão dos chatbots as tornou acessíveis, promovendo sua ampla adoção e integração em plataformas e mecanismos de pesquisa existentes na internet. Isso levanta questões fundamentais: como funciona uma inteligência artificial, na prática? Qual é a ciência que fundamenta esses sistemas que, muitas vezes, parecem realizar feitos quase mágicos em nosso cotidiano, desde auxiliar na escrita até resolver equações complexas?

Este trabalho propõe investigar os mecanismos subjacentes às redes neurais, explorando a matemática envolvida nesses algoritmos e como ela se relaciona com o currículo do ensino médio, alinhado às diretrizes da nova estrutura educacional.

Prosseguindo com a exploração do tema, é importante destacar que a matemática que

fundamenta as redes neurais está profundamente interligada com conceitos fundamentais da educação básica. Por exemplo, o cálculo diferencial e integral, que desempenha um papel crucial no ajuste dos pesos sinápticos durante o processo de aprendizado da rede, é um tópico que pode ser introduzido no ensino médio.

Além disso, a Álgebra Linear, com suas matrizes e vetores, é essencial para entender como as informações são processadas e transmitidas através das camadas da rede. Estes são conceitos que, quando ensinados em contexto, podem não apenas enriquecer o currículo, mas também preparar os estudantes para um mundo cada vez mais dominado pela tecnologia e pela análise de dados.

A inteligência artificial e, por extensão, as redes neurais, estão remodelando o panorama de várias indústrias, desde a medicina, com diagnósticos mais precisos e personalizados, até a agricultura, com sistemas de monitoramento e controle de cultivos mais eficientes. Isso sem mencionar o impacto na economia, na logística e no entretenimento. Portanto, compreender essas tecnologias não é apenas uma questão de curiosidade acadêmica, mas uma necessidade prática e urgente.

Nesse sentido, a educação tem o papel de não apenas informar, mas também de capacitar os jovens a serem criadores e não apenas consumidores dessas tecnologias. Ao integrar a matemática das redes neurais ao currículo, podemos inspirar uma nova geração de cientistas, engenheiros e empreendedores que irão moldar o futuro da inteligência artificial e contribuir para o avanço da sociedade.

Avançando ainda mais na discussão, é essencial reconhecer que a interseção entre a inteligência artificial e a educação vai além do ensino de conceitos matemáticos. Ela abre portas para uma abordagem interdisciplinar que pode transformar o aprendizado. Ao incorporar exemplos práticos de redes neurais no currículo, os estudantes podem ver a aplicação direta da teoria na resolução de problemas reais, incentivando o pensamento crítico e a inovação.

A inteligência artificial também pode ser uma ferramenta poderosa no próprio processo educacional, personalizando o aprendizado para atender às necessidades individuais dos alunos, identificando pontos fortes e áreas que necessitam de reforço. Isso pode ser alcançado via sistemas adaptativos que ajustam o conteúdo e o ritmo com base no desempenho do aluno, proporcionando uma experiência de aprendizado mais eficaz e envolvente.

Além disso, a familiaridade com as redes neurais prepara os estudantes para os desafios do futuro mercado de trabalho, onde a capacidade de trabalhar lado a lado com a inteligência artificial será uma habilidade valiosa. Eles não só entenderão melhor como essas tecnologias funcionam, mas também estarão equipados para contribuir para o seu desenvolvimento e aplicação em uma variedade de campos.

Portanto, ao explorar a matemática e a ciência por trás das redes neurais, este trabalho não apenas lança luz sobre os princípios fundamentais da inteligência artificial, mas também

visa inspirar uma abordagem educacional que prepare os jovens para se tornarem inovadores e líderes na era da informação. É uma jornada que começa na sala de aula, mas se estende muito além dela, buscando moldar o futuro da tecnologia e da sociedade.

2 REDES NEURAIS

2.1 O desenvolvimento dos sistemas de IA nas últimas décadas

As redes neurais artificiais são estudadas desde o século XX porém, o assunto que era do mundo acadêmico se tornou popular na segunda década do século XXI. Neste período alguns acontecimentos foram muito importantes, que são eles:

- Em 2015, um sistema de IA chamado "Eugene Goostman"conseguiu convencer 33% dos juízes humanos de que se tratava de um menino ucraniano de 13 anos, tornando-se a primeira máquina a passar no Teste de Turing. (WARWICK; SHAH, 2016)
- Em 2017, pesquisadores do Google desenvolveram um sistema de IA chamado "DeepMind"que foi capaz de aprender a jogar Go em um nível sobre-humano, analisando milhares de jogos anteriores jogados por especialistas humanos. (SILVER et al., 2016)
- Também em 2017, uma equipe de pesquisadores da OpenAI desenvolveu um sistema de IA chamado "Dactyl", capaz de aprender sozinho a manipular objetos em um ambiente virtual usando uma mão robótica, escrito por (ANDRYCHOWICZ et al., 2018)
- Em 2018, o 'Pluribus', um sistema de IA desenvolvido pelo Facebook e pela Carnegie Mellon University, derrotou jogadores profissionais em um jogo multiplayer de Texas Hold'em sem limite. A tomada de decisões estratégicas e a adaptabilidade da Pluribus demonstraram o potencial da IA em domínios complexos e de informação imperfeita, escrito por (BROWN; SANDHOLM, 2019)
- Em 2020, um sistema de IA chamado "Benjamin"foi desenvolvido por pesquisadores do Instituto de Tecnologia de Massachusetts para escrever e produzir um curta-metragem chamado "Sunspring". O filme, que foi escrito inteiramente pelo sistema de IA, foi descrito por um crítico como "inacessível", feito por (BENJAMIN, 2016).

Mas afinal, como é ou desenvolvido um sistema de inteligência artificial? Quais os conceitos matemáticos presentes no desenvolvimento das inteligências artificiais? Como uma inteligência artificial elabora uma resposta? E qual é o grau de confiança desta resposta?

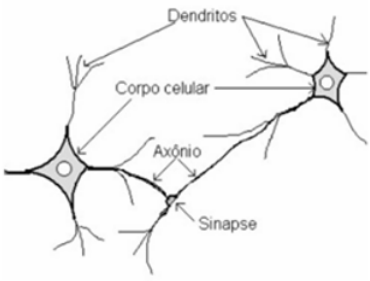
A investigação dessas respostas nos levou a explorar os algoritmos de redes neurais. As seções subsequentes abordarão a teorização e a aplicação prática desses sistemas.

2.2 Os algoritmos de redes neurais

As redes neurais artificiais (RNAs) são sistemas computacionais que imitam o processamento de informações do cérebro humano, simulando o processo de funcionamento de um

neurônio biológico, como indica a Figura 1.

Figura 1 – Neurônio biológico.

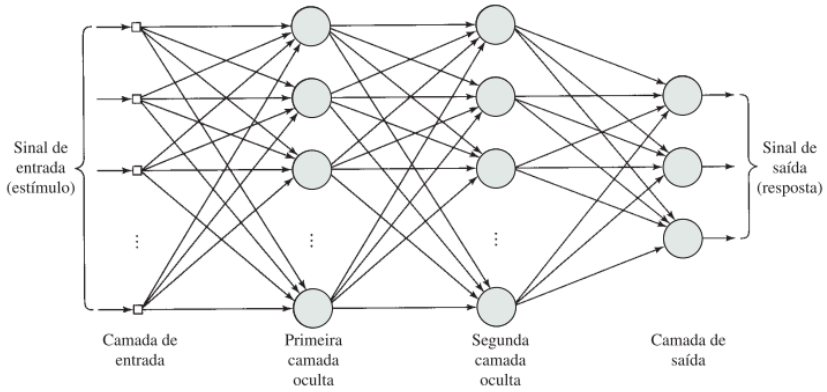


Fonte: (NETO; BONINI, 2010, p.88)

Os neurônios são células especializadas do sistema nervoso que transmitem informações mediante sinais elétricos e químicos. Eles recebem estímulos externos ou de outros neurônios pelos dendritos. Se o estímulo for forte o suficiente, o neurônio gera um impulso elétrico que viaja pelo axônio até chegar às terminações nervosas. As substâncias químicas chamadas neurotransmissores são liberadas e atravessam a sinapse, o espaço entre os neurônios, para transmitir o sinal ao próximo neurônio ou célula-alvo. Este processo permite a comunicação rápida e eficiente dentro do sistema nervoso, essencial para todas as funções do corpo e atividades mentais.

As RNAs são formadas por unidades de processamento simples, chamadas neurônios artificiais, que se conectam através de sinapses artificiais. Essas conexões formam uma estrutura semelhante a um grafo[1], onde os nós representam os neurônios e as arestas, as sinapses.

Figura 2 – Grafo de uma rede perceptron de múltiplas camadas



Fonte: (BRAGA; CARVALHO; LUDEMIR, 2000, p.186)

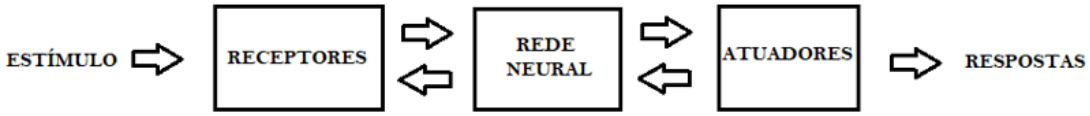
As redes neurais desempenham várias funções em diferentes setores, como: diagnóstico médico feito pela classificação de imagens, marketing direcionado pela filtragem de mídia social e análise de dados comportamentais, previsões financeiras feitas pelo processamento de

[1] Um grafo é uma estrutura matemática composta por vértices (ou nós) e arestas (ou arcos) que representam as relações entre esses vértices

dados históricos de instrumentos financeiros, previsões de demanda de energia e carga elétrica, processo e controle de qualidade, identificação de compostos químicos, dentre outros.

A estrutura de uma rede neural é representada por técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Como indica o esquema abaixo:

Figura 3 – Representação em diagrama de blocos do sistema nervoso



Fonte: Elaborado por (HAYKIN, 2007, p.32)

Uma das propriedades mais importantes de uma rede neural artificial é a capacidade de aprender por intermédio de exemplos e fazer inferências sobre o que aprendeu, melhorando gradativamente o seu desempenho. As redes neurais utilizam um algoritmo de aprendizagem cuja tarefa é ajustar os pesos de suas conexões (BRAGA; CARVALHO; LUDEMIR, 2000, cap.2).

Em um neurônio biológico, a informação recebida pelo ser humano é ponderada por um valor específico. Por exemplo, ao nos depararmos com uma chama, evitamos tocá-la, uma vez que já sabemos que ela queima. No entanto, uma criança que encontra uma chama pela primeira vez talvez se incline a tocá-la, recebendo um estímulo negativo como resposta. Isso ocorre porque ela ainda não desenvolveu as conexões neurais necessárias para associar a visão do fogo com o perigo potencial, uma resposta aprendida com a experiência.

2.3 A estrutura de uma rede neural

A estrutura de uma rede neural artificial é uma organização hierárquica de várias camadas de neurônios artificiais que trabalham juntos para aprender a partir de dados. Cada camada é composta por um conjunto de neurônios artificiais que processam as entradas e geram saídas para a próxima camada. As camadas são geralmente divididas em três tipos principais:

- Camada de Entrada: onde os padrões são apresentados à rede;
- Camadas Intermediárias ou Escondidas: onde é feita a maioria do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
- Camada de Saída: onde o resultado é concluído e apresentado.

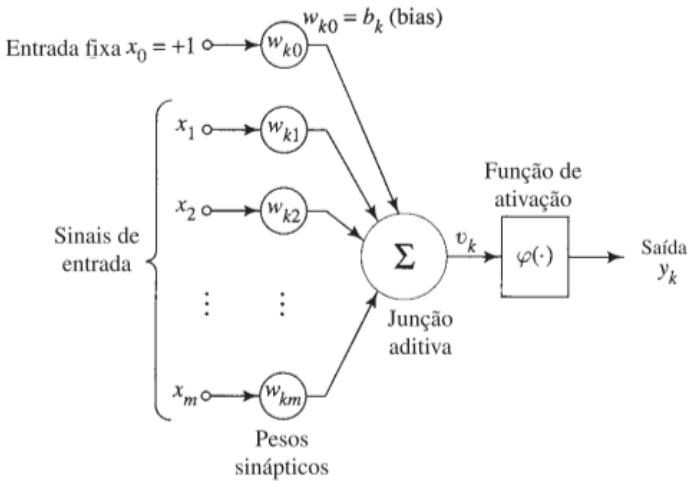
Cada camada da rede neural é conectada à camada seguinte por uma matriz de pesos que determina como as entradas são transformadas em saídas para a próxima camada da rede.

Durante o treinamento da rede neural, esses pesos são ajustados iterativamente para minimizar a diferença entre os resultados previstos e os resultados reais. A estrutura de camadas da rede neural é ajustada para cada problema específico, e geralmente consiste em um número variável de camadas e neurônios.

As redes neurais artificiais se diferenciam pela sua arquitetura e pela forma como os pesos associados às conexões são ajustados durante o processo de aprendizado. A arquitetura de uma rede neural restringe o tipo de problema no qual a rede poderá ser utilizada, e é definida pelo número de camadas (camada única ou múltiplas camadas), pelo número de nós em cada camada, pelo tipo de conexão entre os nós (feedforward ou feedback) e por sua topologia (HAYKIN, 2007, p.46-49).

Na Figura 4, temos a representação de um neurônio artificial, cada dado de entrada (x_i) é associado a um peso (w_i) e a uma variável de controle (*bias*[2]) b_i e processado por uma função de ativação gerando um valor de saída $y = \sigma(z)$ para o próximo neurônio. De forma geral, cada neurônio de uma rede neural pode ser representado pelo esquema da Figura 4:

Figura 4 – Neurônio artificial.



Fonte: Elaborado por (HAYKIN, 2007, p.38)

A equação abaixo, corresponde a representação matemática do esquema apresentado na Figura 4, sendo n o número de neurônios presentes em uma determinada camada da rede neural:

$$z = x_1\ w_1 + x_2\ w_2 + \ldots + x_n\ w_n + b_n = \sum_{i=1}^{n} x_i w_i + b_i \tag{1}$$

A variável de controle *bias* é acrescentada na soma ponderada z dos pesos com os vieses, para fazer com que a presença do zero nos pesos ou nos neurônios não faça com que toda a rede neural convirja para zero. A soma ponderada (z) representa a iteração de um neurônio específico da próxima camada com os neurônios e *bias* da camada anterior da rede. Conside-

[2] A tradução da palavra *bias* é viés, porém adotaremos a palavra *bias* para a variável de controle neste trabalho

rando que a próxima camada pode conter j neurônios, logo teremos j somas ponderadas em uma determinada camada, e esta ideia é válida para qualquer camada da rede.

As bibliografias sobre redes neurais utilizam o índice sobrescrito entre parenteses $z^{(L)}$ para representar as camadas das RNAs, diferente da ideia matemática que o utiliza como expoente. Neste trabalho iremos utilizá-lo para representar as camadas, porém sem os parenteses em cada índice sobrescrito, logo temos que $L = 0$ corresponde aos valores de entrada, $L = 1$ à primeira camada oculta e assim por diante, temos também a camada sucessora de L como $L+1$ e a camada anterior $L-1$. Com isso, a equação que descreve a soma ponderada para um neurônio j em qualquer camada L pode ser generalizada da seguinte forma:

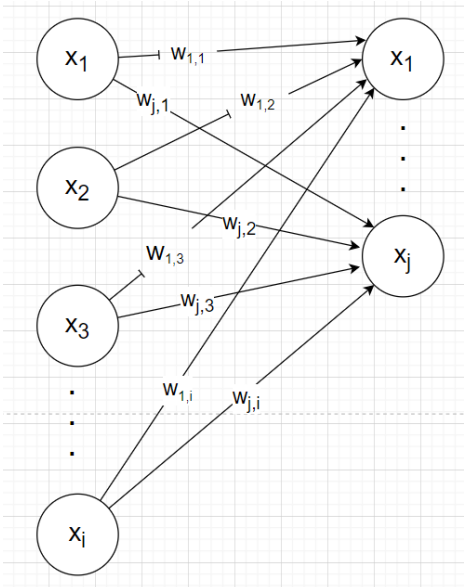
$$z_j^L = \sum_{i=1}^{n} w_{ji}^L \cdot x_i^{L-1} + b_j^L \tag{2}$$

na qual L representa a camada, j representa um índice de um neurônio na camada L, z representa a soma ponderada dos pesos com os neurônios x da camada anterior $L-1$ e i representa o índice de um neurônio na camada $L-1$.

2.3.1 Matriz de pesos e *bias*

A equação 2 também pode ser generalizada para os vários neurônios de uma rede perceptron de múltiplas[3] camadas escrita na forma matricial. Considere a imagem abaixo que representa a relação de i neurônios de uma camada com j neurônios na próxima camada:

Figura 5 – Relação entre neurônios em uma rede perceptron de múltiplas camadas.



Fonte: elaborado pelo autor

[3] Um perceptron de múltiplas camadas é uma rede neural artificial composta por várias camadas de neurônios, que utilizam funções de ativação não lineares para aprender padrões complexos nos dados

A Figura 5 ilustra a relação entre os pesos e os neurônios de duas camadas quaisquer da rede neural, na qual W^L é a matriz de pesos que conecta todos os neurônios de uma camada L com os neurônios da camada anterior, $L-1$. Desta forma, cada componente da matriz W^L, representa um peso de coordenadas, $w_{j,i}$, que fará o papel de ligar os neurônios de duas camadas quaisquer da rede neural, onde as linhas da matriz, representada pelo índice j, corresponde a posição de um neurônio na camada L e as colunas, os índices i, representam os neurônios da camada anterior $L-1$. Assim, podemos representar a matriz de pesos de uma camada qualquer, como sendo:

$$W^L = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,i} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,i} \\ \vdots & \ddots & \cdots & \vdots \\ w_{j,1} & w_{j,2} & \cdots & w_{j,i} \end{bmatrix} \tag{3}$$

Cada neurônio de uma determinada camada L receberá um único *bias*, como variável de controle. Desta forma, teremos para cada camada um vetor coluna B^L, definido como:

$$B^L = \begin{bmatrix} b_1^L \\ b_2^L \\ \vdots \\ b_j^L \end{bmatrix} \tag{4}$$

Dessa forma, a equação 2, que representa a soma ponderada z^L para todos os neurônios de uma determinada camada L, pode ser escrita da seguinte maneira:

$$\begin{bmatrix} z_1^L \\ z_1^L \\ \vdots \\ z_j^L \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,i} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,i} \\ \vdots & \ddots & \cdots & \vdots \\ w_{j,1} & w_{j,2} & \cdots & w_{j,i} \end{bmatrix} \begin{bmatrix} x_1^{L-1} \\ x_2^{L-1} \\ \vdots \\ x_i^{L-1} \end{bmatrix} + \begin{bmatrix} b_1^L \\ b_2^L \\ \vdots \\ b_j^L \end{bmatrix} \tag{5}$$

Ou ainda,

$$Z^L = W^L X^L + B^L$$

A participação de um aluno em aulas ou grupos de estudo não garante a assimilação do conteúdo proposto, evidenciando que a aprendizagem não é um evento certo, mas sim uma probabilidade. Em nossa rede neural, a função de ativação é responsável por transformar a soma ponderada em um valor entre 0 e 1. Na próxima seção, apresentaremos um resumo sobre a função de ativação utilizada neste estudo.

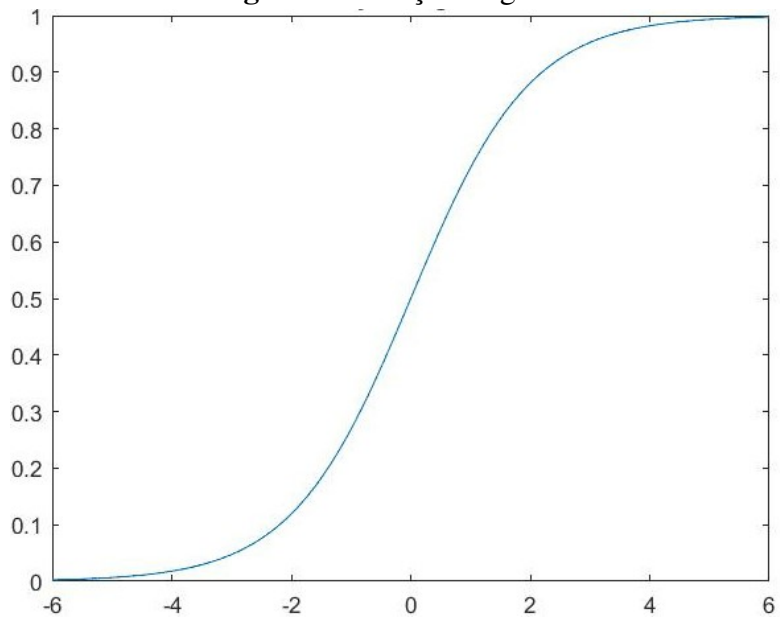
2.3.2 Função de ativação

No treinamento de algoritmos de redes neurais existem diversas funções de ativação cujo objetivo é definir a saída enviada para o próximo neurônio. Neste trabalho utilizaremos uma classe de funções definidas como funções de tipo sigmoide, a mais conhecida entre elas é a função logística que, transforma o número de saída z em um número que está no intervalo entre 0 e 1.

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{6}$$

A função também pode ser representada pelo gráfico abaixo:

Figura 6 – Função Logística.



Fonte: elaborado pelo autor

Desta forma, o estado de cada neurônio de uma camada L com i neurônios em uma rede perceptron de múltiplas camadas, pode ser calculado da seguinte forma:

$$x_i^L = \sigma(z_i^L) \tag{7}$$

E para os valores de saída na camada L, temos j neurônios da próxima camada $L+1$ apresentados como x_j^{L+1}, logo teremos:

$$x_j^{L+1} = \sigma\left(\sum_{i=1}^{n} x_i^L w_{j,i}^{L+1} + b_j^{L+1}\right) = \sigma(z_j^{L+1}) \tag{8}$$

Uma vez definidas as interações para uma camada específica, nosso principal desafio é determinar os valores para $w_{j,i}^{L+1}$ e b_j^{L+1}, de tal forma que os valores de saída na última camada x_i^L sejam condizentes com os dados utilizados para treinar a rede neural.

2.3.3 Custo da rede neural e o método *backpropagation*

Na primeira iteração de uma rede neural, os pesos $w_{j,i}$ e os valores de ajuste fino (*bias*) são definidos aleatoriamente. Por causa disso, o erro inicial da rede, também conhecido como custo, tende a ser elevado quando novos dados são processados. Para melhorar o desempenho da rede, utilizamos métodos de treinamento como os algoritmos de aprendizado supervisionado. Nesse método, as saídas geradas pela rede são comparadas com um conjunto de dados de referência y_i, o que nos permite medir e corrigir os erros através da equação:

$$E = \sum_{i=1}^{n} (x_i^L - y_i)^2 \tag{9}$$

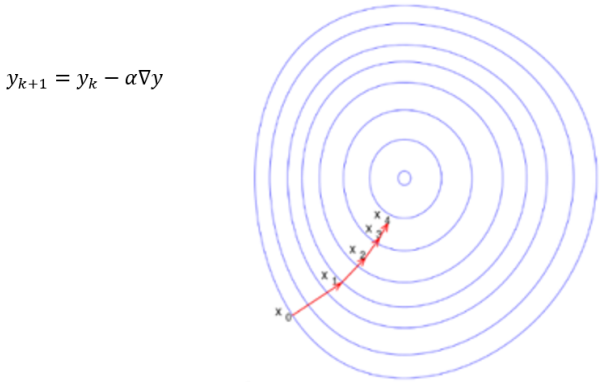
Assim, para cada iteração feita por nossa rede, podemos comparar os resultados gerados, com o que existe no banco de dados utilizando a soma dos quadrados das diferenças para definir o quanto a nossa rede está errando. Essa soma nos dá uma medida quantitativa do desempenho da rede neural, permitindo ajustes nos pesos e *bias* para melhorar a precisão nas previsões subsequentes.

As bibliografias da área, chamam a soma dos erros ao quadrado de função custo da rede, logo adotaremos:

$$C = \sum_{i=1}^{n} (x_i^L - y_i)^2 \tag{10}$$

Definida a função custo da rede, o nosso problema central, consiste em fazer com que este custo se aproxime de 0. A forma mais utilizada para minimizar a função custo é o método de máxima descida ou contra gradiente. A ideia é calcular o vetor gradiente da função sigmoide, a partir daí, tentamos mover os parâmetros na direção contra-gradiente $(-\nabla C)$, que é a direção de maior decréscimo local da função C.

Figura 7 – Método do contra gradiente.



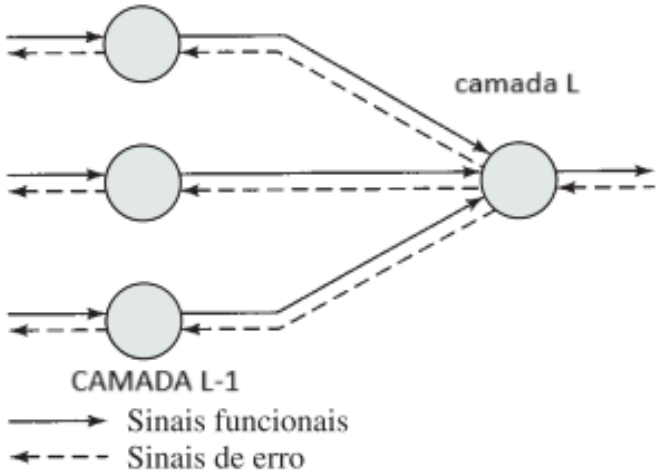
Fonte:(GRADIENT, 2024)

A utilização do gradiente descendente é aplicado na rede neural, a partir do método retropropagação (*backpropagation*). O termo retropropagação é a tradução literal de *backpro-*

pagation, porém as bibliografias da área sempre utilizam o termo *backpropagation* no lugar de retropropagação.

A ideia deste treinamento é apresentar um dado à camada de entrada da rede (sinais funcionais), e atribuir pequenos valores para os pesos, processá-los camada após camada e, a partir do valor de saída, verificar o erro. Busca-se reduzir o erro, calculando o contra gradiente a partir da camada de saída (Sinais de erro) até a camada de entrada, corrigindo os pesos a partir da saída até chegar na camada de entrada. Desta forma, repetimos o processo até que o erro da rede seja o mínimo possível ou aceitável.

Figura 8 – Grafo ilustrado o método *backpropagation*.



Fonte: (BRAGA; CARVALHO; LUDEMIR, 2000, p.186)

O vetor gradiente é calculado pelas derivadas parciais $\frac{\partial C}{\partial w_{ij}^L}$ e $\frac{\partial C}{\partial b_i^L}$, onde este processo é calculado de trás para frente na rede neural, começando pela última camada, em que $L = 3$ até chegar na primeira camada da rede neural para $L = 1$. Desta forma, os pesos são ajustados na direção oposta ao gradiente e o processo é repetido para múltiplas interações até que o erro total da rede seja mínimo. Este processo permite que a rede aprenda padrões complexos e ajuste seus pesos internos para realizar tarefas como classificação e regressão.

No próximo capítulo, exploraremos o algoritmo de redes neurais, definindo as derivadas parciais para ajustar os pesos e *bias*, com o objetivo de resolver um problema de reconhecimento de números manuscritos entre 0 e 9.

3 O ALGORITMO DA REDE NEURAL PARA RECONHECIMENTO DE IMAGENS

O reconhecimento de imagem é uma tarefa desafiadora que envolve identificar objetos, padrões ou características em imagens. Neste capítulo, exploraremos o algoritmo de reconhecimento de imagem usando redes neurais, com foco no conjunto de dados MNIST [1] da Microsoft.

O conjunto de dados MNIST é um marco na área de visão computacional. Ele consiste em 70.000, onde 60.000 são para treinamento da rede neural e 10.000 imagens para teste, imagens de dígitos manuscritos (de 0 a 9), cada uma com 28×28 pixeis. Essas imagens são usadas para treinar e testar modelos de reconhecimento de dígitos. O problema de reconhecimento de algarismos é frequentemente considerado o "Hello World" dos algoritmos de aprendizado de máquina.

3.1 O problema

As redes neurais artificiais são ferramentas poderosas para o reconhecimento de padrões em imagens, empregando algoritmos de aprendizado profundo para essa tarefa. Elas são treinadas com um conjunto diversificado de imagens rotuladas, onde cada rótulo indica uma categoria distinta, como números, animais ou objetos. Durante o treinamento, a rede ajusta os pesos sinápticos para aprimorar sua habilidade de diferenciar essas categorias.

Quando uma imagem é apresentada à rede, ela é processada em camadas sucessivas, com cada camada aplicando transformações matemáticas para extrair características cada vez mais complexas da imagem. Essas características são então comparadas com as informações armazenadas na rede durante o treinamento para determinar a categoria provável da imagem.

Uma rede neural pode ser treinada para reconhecimento de dígitos usando um conjunto de dados de imagens. O treinamento da rede implica em alimentar a rede com exemplos e ajustar os pesos das conexões entre os neurônios para minimizar o custo da rede. Uma vez treinada, a rede pode ser usada para prever os dígitos correspondentes a uma imagem de entrada desconhecida.

Para melhorar o reconhecimento, é possível adicionar técnicas de pré-processamento de imagem, como normalização de contraste, filtragem de ruído e segmentação de imagem. Além disso, é recomendável avaliar diferentes arquiteturas de rede, ajustar os parâmetros de treinamento e validar a precisão da rede em um conjunto de dados de teste separado do conjunto de dados de treinamento. A imagem abaixo ilustra exemplos de imagens do banco de dados do pacote MNIST.

As imagens MNIST são convertidas em escala de cinza e redimensionadas para um tamanho uniforme. Os píxeis são normalizados para valores entre 0 e 1. Cada pixel de uma

[1] Disponível em https://tinyurl.com/4bc89kut/, ultimo acesso em 14 de abril de 2024

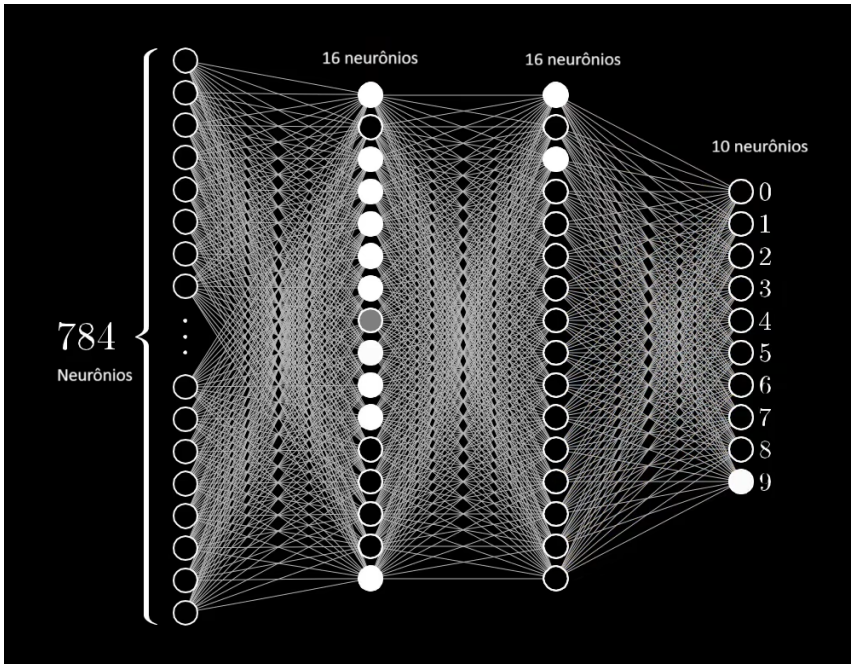
Figura 9 – Reconhecimento de imagens em píxel.

Fonte: (SANDERSON, 2016)

imagem de 28x28 píxeis, tem valores entre 0 e 1 na escala de cinza, sendo que o 0 é para os píxeis pretos e 1 para os píxeis totalmente brancos, cada valor de entrada (píxel) é visto como um neurônio x_n e estes números são o que determina se um neurônio está ou não ativado.

A rede é estruturada com uma camada de entrada de 784 neurônios, correspondendo aos píxeis da imagem, seguida por camadas ocultas com 16 neurônios cada e uma camada de saída com 10 neurônios, representando os dígitos de 0 a 9. A ativação de um único neurônio na camada de saída indica o dígito reconhecido pela rede. A configuração de 16 neurônios nas camadas ocultas segue a recomendação do livro escrito por (NIELSEN M., 2015) "Neural Networks and Deep Learning[2]".

Figura 10 – Processamento em camadas.



Fonte: (SANDERSON, 2016)

Desta forma, cada neurônio da segunda camada recebe os valores de entrada x_n da

[2] Disponível em http://neuralnetworksanddeeplearning.com/

primeira camada multiplicado por um peso w mais um parâmetro de controle chamado *bias* b_n, logo o primeiro neurônio da segunda camada será representado pela soma ponderada:

$$z_1^1 = w_{1,1}^1 \cdot x_1^0 + w_{1,2}^1 \cdot x_2^0 + w_{1,3} \cdot x_3^0 + \cdots + w_{1,784} \cdot x_{784}^0 + b_1^1.$$

Para os 16 neurônios da primeira camada, temos:

$$Z^1 = \begin{bmatrix} w_{1,1}^1 & w_{1,2}^1 & \cdots & w_{1,784}^1 \\ w_{2,1}^1 & w_{2,2}^1 & \cdots & w_{2,784}^1 \\ \vdots & \ddots & \cdots & \vdots \\ w_{16,1}^1 & w_{16,2}^1 & \cdots & w_{16,784}^1 \end{bmatrix} \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_{784}^0 \end{bmatrix} + \begin{bmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_{16}^1 \end{bmatrix} = \begin{bmatrix} z_1^1 \\ z_2^1 \\ \vdots \\ z_{16}^1 \end{bmatrix} \tag{11}$$

Assim, a iteração dos neurônios na camada de entrada, $L = 0$, com os neurônios da primeira camada oculta da rede, geram um vetor coluna com 16 neurônios. As interações da primeira com a segunda camada, podem ser representadas pelo produto dos pesos da segunda camada com o vetor resultante, logo temos a operação matricial abaixo:

$$Z^2 = \begin{bmatrix} w_{1,1}^2 & w_{1,2}^2 & \cdots & w_{1,16}^2 \\ w_{2,1}^2 & w_{2,2}^2 & \cdots & w_{2,16}^2 \\ \vdots & \ddots & \cdots & \vdots \\ w_{16,1}^2 & w_{16,2}^2 & \cdots & w_{16,16}^2 \end{bmatrix} \begin{bmatrix} x_1^1 \\ x_2^1 \\ \vdots \\ x_{16}^1 \end{bmatrix} + \begin{bmatrix} b_1^2 \\ b_2^2 \\ \vdots \\ b_{16}^2 \end{bmatrix} = \begin{bmatrix} z_1^2 \\ z_2^2 \\ \vdots \\ z_{16}^2 \end{bmatrix} \tag{12}$$

E para a camada de saída da rede temos o vetor resultante da iteração entre a primeira e segunda camada, com a matriz de pesos da terceira camada mais os vieses, logo temos como saída a matriz:

$$Z^3 = \begin{bmatrix} w_{1,1}^3 & w_{1,2}^3 & \cdots & w_{1,16}^3 \\ w_{2,1}^3 & w_{2,2}^3 & \cdots & w_{2,16}^3 \\ \vdots & \ddots & \cdots & \vdots \\ w_{10,1}^3 & w_{10,2}^3 & \cdots & w_{10,16}^3 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_{16}^2 \end{bmatrix} + \begin{bmatrix} b_1^3 \\ b_2^3 \\ \vdots \\ b_{10}^3 \end{bmatrix} = \begin{bmatrix} z_1^3 \\ z_2^3 \\ \vdots \\ z_{10}^3 \end{bmatrix} \tag{13}$$

Cada um dos neurônios da rede neural, será ativado aplicando cada uma das somas ponderadas presentes em Z^1, Z^2 e Z^3 na equação 7. Deste forma, a partir de uma iteração da rede, podemos avaliar o custo para os 10 neurônios da camada de saída do seguinte modo:

$$C = \sum_{i=1}^{10} (x_i^3 - y_i)^2 \tag{14}$$

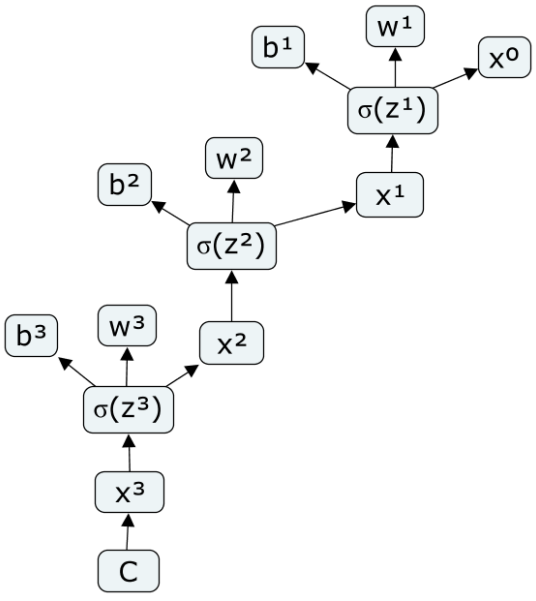
Definido o que é uma iteração para a nossa rede neural, podemos falar sobre o algoritmo de *backpropagation*.

3.2 O algoritmo de backpropagation para a terceira camada da rede neural

O algoritmo de *backpropagation* tem por objetivo alterar os parâmetros da matriz de pesos e *bias* para que as RNAs deem respostas coerentes para os problemas analisados. A fim de minimizar o custo da rede neural, faremos o cálculo do vetor gradiente através das derivadas parciais.

Como a função custo é uma função composta que depende dos valores de saída do neurônio da última camada, e os neurônios da última camada são ativados pela função sigmoide e os valores de saída da função dependem da soma ponderada dos pesos com os neurônios da camada anterior e o *bias*, faremos as derivadas parciais pela regra da cadeia, disponível em (STEWART, 2013, p.831) . O esquema abaixo exemplifica a função composta do custo da rede para a terceira camada da rede neural:

Figura 11 – Relação de dependência da função custo.



Fonte: elaborado pelo autor

Desta forma, temos as derivadas parciais $\frac{\partial C}{\partial b_i^3}$ e $\frac{\partial C}{\partial w_{ij}^3}$. Para a derivada parcial do Custo em relação ao *bias*, começando pelos neurônios da terceira camada da nossa rede, temos:

$$\frac{\partial C}{\partial b_i^3} = \frac{\partial C}{\partial x_i^3} \cdot \frac{\partial x_i^3}{\partial z_i^3} \cdot \frac{\partial z_i^3}{\partial b_i^3}. \tag{15}$$

Como $C = \sum_{i=1}^{10}(x_i^3 - y_i)^2 \Rightarrow \frac{\partial C}{\partial x_i^3} = 2(x_i^3 - y_i^3)$. Temos que $z_i^3 = \sum_{j=1}^{16} x_j^2 w_{i,j}^3 + b_i^3 \Rightarrow \frac{\partial z_i^3}{\partial b_i^3} = 1$. E como $x_i^3 = \sigma(z_i^3) \Rightarrow \frac{\partial x_i^3}{\partial z_i^3} = \sigma'(z_i^3)$. Substituindo os resultados em (15), temos:

$$\frac{\partial C}{\partial b_i^3} = 2(x_i^3 - y_i^3) \cdot \sigma'(z_i^3) \cdot 1. \tag{16}$$

Agora, fazendo a derivada parcial da função custo com relação aos pesos, pela regra da cadeia obtemos:

$$\frac{\partial C}{\partial w_{ij}^3} = \frac{\partial C}{\partial x_i^3} \cdot \frac{\partial x_i^3}{\partial z_i^3} \cdot \frac{\partial z_i^3}{\partial w_{ij}^3}. \tag{17}$$

A derivada da função custo em relação aos neurônios da terceira camada e a variação dos neurônios da terceira camada em relação à soma ponderada z continuam os mesmos da equação (15), o que muda é a variação da soma ponderada em relação aos pesos, que pode ser escrito como:

$z_i^3 = \sum_{j=1}^{n} x_j^2 w_{ij}^3 + b_i^3 \Rightarrow \frac{\partial z_i^3}{\partial w_{ij}^3} = x_j^2.$

Substituindo os resultados na equação (17), obtemos:

$$\frac{\partial C}{\partial w_{ij}^3} = 2(x_i^3 - y_i^3) \cdot \sigma'(z_i^3) \cdot x_j^2. \tag{18}$$

Concluindo assim as derivadas parciais da função custo para os pesos e *bias* para a terceira camada da rede. Logo, cada peso da matriz w_{ij}^3 será atualizado da seguinte forma:

$$w_{ij}^3 \leftarrow w_{ij}^3 - \alpha \cdot \frac{\partial C}{\partial w_{ij}^3} \tag{19}$$

E o *bias* será atualizado:

$$b_i^3 \leftarrow b_i^3 - \alpha \cdot \frac{\partial C}{\partial b_i^3} \tag{20}$$

A constante α é o parâmetro utilizado como taxa de aprendizagem da rede, seu uso é justificado para valores pequenos visando proporcionar uma descida suave no plano. Caso α assuma valores grandes, a convergência segue um caminho oscilatória e quando α excede um certo valor crítico o algoritmo se torna instável, segundo (HAYKIN, 2007, p.148).

3.3 Cálculos de backpropagation para a segunda camada da rede neural

Na segunda camada temos 16 neurônios, que se relacionam com 16 neurônios da primeira camada e 16 neurônios da terceira camada. Um neurônio da segunda camada, recebe as informações de 16 neurônios da primeira camada. Logo, faremos a derivada parciais da função custo em relação aos pesos e *bias* $\frac{\partial C}{\partial w_{ij}^2}$ e $\frac{\partial C}{\partial b_i^2}$ da camada anterior.

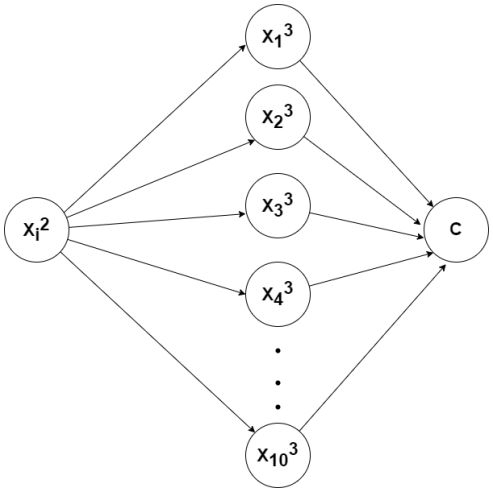
Vamos começar pela derivada da função custo em relação ao *bias*, logo temos que:

$$\frac{\partial C}{\partial b_i^2} = \frac{\partial C}{\partial x_i^2} \cdot \frac{\partial x_i^2}{\partial z_i^2} \cdot \frac{\partial z_i^2}{\partial b_i^2}. \tag{21}$$

Para as derivadas parciais de um neurônio da segunda camada x_i^2 em relação à soma ponderada z_i^2 e a variação de z_i^2 em relação ao *bias* da segunda camada b_i^2 são semelhantes

ao caso anterior. Temos que $\frac{\partial x_i^2}{\partial z_i^2} = \sigma'(z_i^2)$ e $\frac{\partial z_i^2}{\partial b_i^2} = 1$. A mudança acontece para a variação do custo da rede em relação aos neurônios da segunda camada, pois o erro em cada neurônio da segunda camada vai influenciar diretamente no erro de todos os neurônios da terceira camada. A imagem abaixo mostra a iteração de um nerônio da segunda camada com os neurônios da terceira camada:

Figura 12 – Regra da cadeia para cálculos das derivadas parciais do erro na segunda camada.



Fonte: elaborado pelo autor

Aplicando a regra da cadeia, temos:

$$\frac{\partial C}{\partial x_i^2} = \frac{\partial C}{\partial x_1^3} \cdot \frac{\partial x_1^3}{\partial x_i^2} + \frac{\partial C}{\partial x_2^3} \cdot \frac{\partial x_2^3}{\partial x_i^2} + \ldots + \frac{\partial C}{\partial x_{10}^3} \cdot \frac{\partial x_{10}^3}{\partial x_i^2}$$

ou seja

$$\frac{\partial C}{\partial x_i^2} = \sum_{j=1}^{10} \frac{\partial C}{\partial x_j^3} \cdot \frac{\partial x_j^3}{\partial x_i^2}. \tag{22}$$

Temos que:

$$\frac{\partial C}{\partial x_i^3} = 2 \cdot (x_i^3 - y_i). \tag{23}$$

e por definição:

$$x_j^3 = \sigma(w_{j1}^3 \cdot x_1^2 + w_{j2}^3 \cdot x_2^2 + w_{j3} \cdot x_3^3 + \cdots + w_{j16} \cdot x_{16}^2 + b_j^3) = \sigma\left(\sum_{i=1}^{16} x_i^2 w_{ji}^3 + b_i^3\right). \tag{24}$$

Como os neurônios da terceira camada também são formados por uma função composta da sigmoide com o produto dos pesos pelos neurônios da segunda camada, faremos a derivada parcial por regra da cadeia:

$$\frac{\partial x_j^3}{\partial x_i^2} = \frac{\partial x_i^3}{\partial z_i^3} \cdot \frac{\partial z_i^3}{\partial x_j^2}. \tag{25}$$

Da equação (24), temos que $\frac{\partial x_i^3}{\partial z_i^3} = \sigma'(z_i^3)$ e $\frac{\partial z_i^3}{\partial x_j^2} = w_{ji}^3$, substituindo os resultados em (25), obtemos:

$$\frac{\partial x_j^3}{\partial x_i^2} = w_{ji}^3 \cdot \sigma'(z_j^3). \tag{26}$$

Substituindo os resultados encontrados em (23) e (26) em (22), obtemos:

$$\frac{\partial C}{\partial x_i^2} = \sum_{j=1}^{10} 2 \cdot (x_j^3 - y_j) \cdot \sigma'(z_j^3) \cdot w_{ji}^3. \tag{27}$$

Substituindo os resultados encontrados em (21), a derivada do custo em relação ao *bias* da segunda camada:

$$\frac{\partial C}{\partial b_i^2} = \sigma'(z_i^2) \cdot \left[\sum_{j=1}^{10} \sigma'(z_j^3) \cdot w_{ji}^3 \cdot 2 \cdot (x_j^3 - y_j) \right]. \tag{28}$$

A variação do custo com relação aos pesos pode ser obtida através da regra da cadeia:

$$\frac{\partial C}{\partial w_{ij}^2} = \frac{\partial C}{\partial x_i^2} \cdot \frac{\partial x_i^2}{\partial z_i^2} \cdot \frac{\partial z_i^2}{\partial w_{ij}^2}. \tag{29}$$

O que muda em relação à variação do custo pelo *bias* é que $\frac{\partial z_i^2}{\partial w_{ij}^2} = x_j^1$, portanto:

$$\frac{\partial C}{\partial w_{ij}^2} = x_j^1 \cdot \sigma'(z_i^2) \cdot \left[\sum_{j=1}^{10} \sigma'(z^3) \cdot w_{ji}^3 \cdot 2 \cdot (x_j^3 - y_j) \right]. \tag{30}$$

Portanto, temos o cálculo do vetor gradiente para as alterações dos pesos e *bias* da segunda camada da rede neural.

3.4 O algoritmo de backpropagation para a primeira camada da rede neural

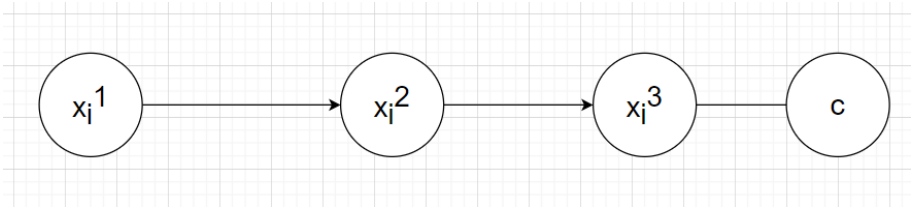
A primeira camada oculta da rede neural se relaciona com os 784 neurônios da camada de entrada da rede neural, e esta camada possui 16 neurônios. Desta forma, calculando o vetor gradiente do custo da rede em relação ao *bias* da primeira camada, utilizando a Regra da Cadeia temos:

$$\frac{\partial C}{\partial b_i^1} = \frac{\partial C}{\partial x_i^1} \cdot \frac{\partial x_i^1}{\partial z_i^1} \cdot \frac{\partial z_i^1}{\partial b_i^1}. \tag{31}$$

Para a derivada parcial dos neurônios da primeira camada em relação à variável z_i^1 e a derivada de z_i^1 em relação ao *bias* não temos alterações em relação ao caso anterior, logo temos que $\frac{\partial x_i^1}{\partial z_i^1} = \sigma'(z_i^1)$ e $\frac{\partial z_i^1}{\partial b_i^1} = 1$.

Para a derivada parcial da função custo em relação aos neurônios da primeira camada, precisamos verificar que o erro nestes neurônios influenciam diretamente o erro dos neurônios da segunda camada que influenciam diretamente o erro nos neurônios da terceira camada, porém a relação entre a função custo e o erro dos neurônios da segunda com a terceira camada foi calculado no passo anterior. O esquema abaixo exemplifica esta dependência:

Figura 13 – Custo da rede na primeira camada.



Fonte: elaborado pelo autor

Desta forma, temos a relação entre a função custo com os neurônios da primeira com a segunda camada da rede neural, logo:

$$\frac{\partial C}{\partial x_i^1} = \frac{\partial C}{\partial x_1^2}\cdot\frac{\partial x_1^2}{\partial x_i^1} + \frac{\partial C}{\partial x_2^2}\cdot\frac{\partial x_2^2}{\partial x_i^1} + \ldots + \frac{\partial C}{\partial x_{16}^2}\cdot\frac{\partial x_{16}^2}{\partial x_i^1} = \frac{\partial C}{\partial x_i^1} = \sum_{j=1}^{16}\frac{\partial C}{\partial x_j^2}\cdot\frac{\partial x_j^2}{\partial x_i^1}$$

As derivadas parciais da função custo em relação a um neurônio da segunda camada e a derivada de um neurônio da segunda camada em relação a um neurônio da primeira camada são funções compostas. Para a segunda relação temos que:

$$z_j^2 = \sum_{k=1}^{16} x_k^1 \cdot w_{jk}^2 + b_j^2$$

e

$$x_j^2 = \sigma(z_j^2)$$

Logo, pela regra da cadeia temos que:

$$\frac{\partial x_j^2}{\partial x_i^1} = \frac{\partial x_j^2}{\partial z_j^2}\cdot\frac{\partial z_j^2}{\partial x_i^1} = \sigma'(z_j^2)\cdot w_{ji}^2$$

E para a derivada do custo em relação aos neurônios da primeira camada temos:

$$\frac{\partial C}{\partial x_i^1} = \sum_{j=1}^{16}\frac{\partial C}{\partial x_j^2}\cdot\frac{\partial x_j^2}{\partial x_i^1} = \sum_{j=1}^{16}\sigma'(z_j^2)\cdot w_{ji}^2\frac{\partial C}{\partial x_j^2},$$

na qual $\frac{\partial C}{\partial x_j^2}$ foi calculado anteriormente.

Portanto, substituindo os resultados encontrados em (31), obtemos que:

$$\frac{\partial C}{\partial b_i^1} = \sigma'(z_i^1) \cdot \frac{\partial C}{\partial x_i^1} \tag{32}$$

A derivada do custo da função em relação aos pesos da primeira camada da rede neural é dada pela regra da cadeia:

$$\frac{\partial C}{\partial w_{ij}^1} = \frac{\partial C}{\partial x_i^1} \cdot \frac{\partial x_i^1}{\partial z_i^1} \cdot \frac{\partial z_i^1}{\partial w_{ij}^1} = \frac{\partial C}{\partial x_i^1} \cdot \sigma'(z_i^1) \cdot x_j^0. \tag{33}$$

Uma vez que $\frac{\partial C}{\partial x_j^2}$ e $\frac{\partial C}{\partial x_j^1}$ são conhecidas, obtemos todas as derivadas parciais de C com relação aos parâmetros, concluindo assim as derivadas parciais para o algoritmo de *backpropagation*. A implementação do algoritmo se dará pelo software Matlab.

3.5 A programação em Matlab/Octave

A implementação de uma rede neural pode ser feita em diversas linguagens de programação, como Python, R, Java, C++ e Matlab e outras. Neste trabalho utilizaremos o Matlab versão 2018 para implementar o algoritmo porém, o código funciona nas versões mais recentes do *software* Octave.

Vamos descrever os passos utilizados para a implementação do nosso algoritmo de rede neural no Matlab. O algoritmo consiste em:

1. Importar as imagens do banco de dados MNIST e transformá-las em matrizes 28×28 na escala de cinza.
2. Transformar os rótulos em vetores com 10 coordenadas.
3. Separar a base de dados de 60.000 imagens em 100 pacotes com 600 imagens.
4. Construir as matrizes de pesos e bias para cada camada da rede neural.
5. Construir a função de ativação.
6. Repetir os processos de i=1 até o número de épocas(ciclo de treinamento). Cada época envolve o processamento de todos os pacotes de imagens.
7. Processar um pacote de imagens, calcule o erro e faça os processos de *backpropation*.
8. Atualiza os pesos e bias e repete o processo para o próximo pacote.

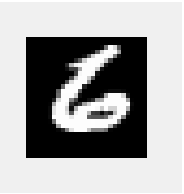
O primeiro passo para a implementação do código no Matlab é fazer o *download* dos pacotes de imagens no site da Microsoft[3]. No site, temos o arquivo 'train-images.idx3-ubyte'

[3] Disponível em https://tinyurl.com/4bc89kut/, ultimo acesso em 14 de abril de 2024

que contém 60.000 imagens para o treino da rede neural e o arquivo 'train-labels.idx1-ubyte' composto pelos rótulos de cada imagem do pacote de treino. Está disponível também, os arquivos 't10k-images.idx3-ubyte' e 't10k-labels.idx1-ubyte' que são os arquivos de imagem e rótulos para teste da rede neural.

O pacote de imagens de treinamento da rede neural não se encontra na forma de vetor, o que temos são 60.000 arquivos de imagens de $28x28$ píxeis, como indicado na Figura 14 abaixo:

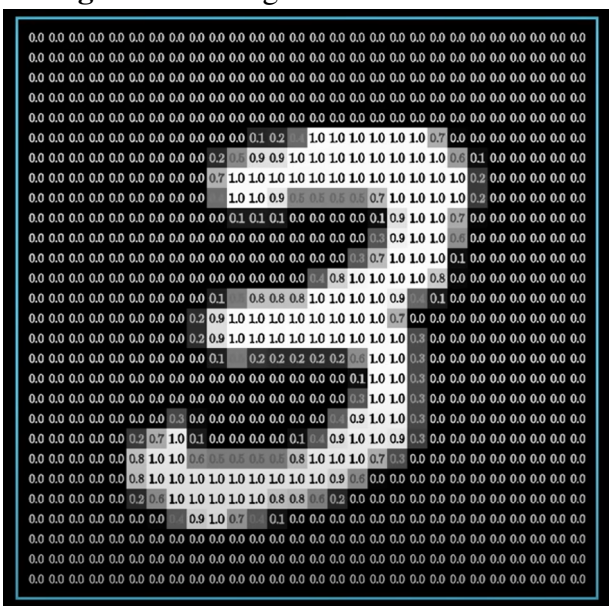
Figura 14 – Imagem do pacote MNIST.



Fonte: (SHUBHIRAJ; FRANKS; LU, 2023)

As imagens processadas pela rede neural precisam ser transformadas em um vetor com 784 coordenadas para ser processado pela rede neural, desta forma precisamos transformar cada pixel em uma escala de cinza, onde 0 representa o pixel que não foi aceso e 1 os píxeis totalmente acesos. A Figura 15 representa esta ideia:

Figura 15 – Imagem na escala de cinza.



Fonte: (SANDERSON, 2016)

O Matlab não fornece nenhuma biblioteca que transforma os dígitos manuscritos do pacote de imagens do MNIST em uma matriz de números da escala de cinza, porém muitos projetos que trabalham com este pacote utilizam a função loadMNISTImages para carregar as imagens do conjunto de dados. Os códigos para esta função são amplamente divulgados em fóruns do Matlab, o código está representado no Quadro 1:

Quadro 1 – Função loadMNISTImages.

```
function images = loadMNISTImages(filename)
fp = fopen(filename, 'rb');
assert(fp ~= -1, ['Could not open ', filename, '']);
magic = fread(fp, 1, 'int32', 0, 'ieee-be');
assert(magic == 2051, ['Bad magic number in ', filename, '']);
numImages = fread(fp, 1, 'int32', 0, 'ieee-be');
numRows = fread(fp, 1, 'int32', 0, 'ieee-be');
numCols = fread(fp, 1, 'int32', 0, 'ieee-be');
images = fread(fp, inf, 'unsigned char');
images = reshape(images, numCols, numRows, numImages);
images = permute(images,[2 1 3]);
fclose(fp);
images = reshape(images, size(images, 1) * size(images, 2),
   size(images, 3));
images = double(images) / 255;
end
```

Fonte: elaborado pelo autor

Cada imagem no conjunto de teste é associada a um rótulo único, armazenado no arquivo train-labels.idx1-ubyte. Por exemplo, a imagem correspondente ao dígito 6 possui o rótulo do numeral 6. No entanto, para facilitar o cálculo do erro na saída da rede neural, é necessário converter esse rótulo em um vetor de 10 elementos. Nesse vetor, cada posição representa um dos possíveis dígitos, de 0 a 9. Portanto, o rótulo do dígito 6 é convertido no vetor (0,0,0,0,0,0,1,0,0,0), onde o '1' está na sétima posição, indicando o número 6. Esse processo é repetido para todos os dígitos, criando uma representação binária que a rede neural pode usar para avaliar sua precisão

A função que faz a leitura dos rótulos é a loadMINISTLabels, que é apresentada no Quadro 2. A função que faz a transformação desses dados em vetores é apresentada no Quadro 3.

Quadro 2 – Função loadMNISTLabels.

```
function labels = loadMNISTLabels(filename)
fp = fopen(filename, 'rb');
assert(fp ~= -1, ['Could not open ', filename, '']);
magic = fread(fp, 1, 'int32', 0, 'ieee-be');
assert(magic == 2049, ['Bad magic number in ', filename, '']);
numLabels = fread(fp, 1, 'int32', 0, 'ieee-be');
labels = fread(fp, inf, 'unsigned char');
assert(size(labels,1) == numLabels, 'Mismatch in label count');
fclose(fp);
end
```

Fonte: elaborado pelo autor

Com os dados prontos, o próximo passo é a construção das matrizes de pesos e *bias* utilizadas pela rede neural. Sabemos que os neurônios de entrada serão fornecidos pela imagem, e estes serão multiplicados por uma matriz de pesos.

Quadro 3 – Função que transforma rótulos em vetor.

```
function Y = labels_vetor (labels)
n=length(labels);% vendo quantos rotulos existem
% criando vetores correspondentes
Y=zeros(10,n);
for i=1:n
  logico=0;
  k=1;
  while logico==0
    if labels(i)==mod(k,10)
      Y(k,i)=1;
      logico=1;
    else
      k=k+1;
    end
    end
  end
end
```

Fonte: elaborado pelo autor

A quantidade de píxeis em uma imagem de 28×28 é de 784 píxeis, onde cada píxel é interpretado como um neurônio de entrada, representado por uma matriz de 784×1 elementos. Esta matriz, contendo os valores de entrada de uma imagem, será multiplicada por outra matriz de pesos, que possui 16×784 elementos, e adicionada a uma matriz de *bias* de 16×1 elementos. Assim, temos uma camada de entrada L0 com 784 elementos, uma camada L1 com 16 elementos, uma camada L2 com 16 elementos e uma última camada L3 com 10 elementos.

Na primeira iteração os pesos e *bias* serão definidos de forma aleatório, como podemos observar no código do Quadro 4.

Quadro 4 – Construção das matrizes de pesos e *bias*.

```
clear all
close all
images=loadMNISTImages('train-images.idx3-ubyte');
labels_dados = loadMNISTLabels('train-labels.idx1-ubyte');
Y_dados=labels_vetor(labels_dados);
L0=784;
L1=16;
L2=16;
L3=10;
W1=randn(L1,L0);
B1=randn(L1,1);
W2=randn(L2,L1);
B2=randn(L2,1);
W3=randn(L3,L2);
B3=randn(L3,1);
```

Fonte: elaborado pelo autor

O próximo passo consiste na divisão do pacote treino com 60.000 imagens em pacotes menores, visando a verificação do erro em pequenos pacotes com 100 imagens. O Matlab possui uma função chamada mat2cell utilizada para converter uma matriz em uma matriz de

células, onde cada célula contém uma submatriz da matriz original. Desta forma, vamos dividir a matriz de imagens de 784×60000 em 600 matrizes de 100 imagens. O mesmo também será feita para o pacote de rótulos das imagens, como indica o Quadro 5.

Quadro 5 – Construção dos pacotes com 100 imagens.

```
24 colunas_por_pacotes = 100;
25 tamanho_pacote = repmat(colunas_por_pacotes,1, size(images,2)/
      colunas_por_pacotes);
26 Y_dados_tamanho_pacote = repmat(colunas_por_pacotes,1,size(
      Y_dados,2)/colunas_por_pacotes);
27 pacotes = mat2cell(images, size(images,1), tamanho_pacote);
28 Y_dados_pacotes = mat2cell(Y_dados, size(Y_dados,1),
      tamanho_pacote);
```

Fonte: elaborado pelo autor

Com os pacotes de imagens bem definidos e as matrizes de pesos e *bias* construídas, ainda precisamos da função de ativação para fazer a primeira iteração da nossa rede neural. A função logística está disponível na biblioteca do Matlab desde a versão R2019d, como estamos utilizando uma versão R2018a do *software*, precisamos chamá-la de um arquivo separado. No Quadro 6, apresentamos a definição da função logística no Matlab.

Quadro 6 – Função de ativação dos neurônios.

```
1 function [w] = sigmoide(x)
2 w = (1)./(1+exp(-x));
3 end
```

Fonte: elaborado pelo autor

Na fase inicial de treinamento da rede neural, enfrentamos o desafio de processar 60.000 imagens, estruturadas em 600 lotes, cada um contendo 100 imagens. Para gerenciar esse volume de dados de forma eficaz, estabelecemos duas variáveis: $t = 600$, que representa o total de lotes, e $x = 100$, que denota a quantidade de imagens em cada lote.

Para assegurar que cada imagem seja submetida ao processo de aprendizado, implementaremos uma estrutura de dois *loops* aninhados. O *loop* externo percorrerá os 600 lotes, enquanto o *loop* interno iterará sobre as 100 imagens dentro de cada lote, aplicando as matrizes de pesos e *bias* mais a função de ativação para cada etapa.

Além disso, para reforçar o aprendizado, um terceiro *loop* será introduzido, permitindo que a rede neural revise o conjunto de treinamento múltiplas vezes. Cada passagem completa pelo conjunto de treinamento é conhecida como uma época. A inclusão desse *loop* adicional é crucial para aprimorar a precisão da rede neural ao longo do tempo, no Quadro 7 é feito 5 iterações, indicado pela variável "epoca".

O pseudocódigo do Quadro 7 ilustra essa iteração:

Esse método iterativo garante uma cobertura abrangente do conjunto de dados e promove uma aprendizagem eficiente para a rede neural.

Quadro 7 – Implementação das matrizes 10, 11 e 12 no *software* Matlab.

```
t = 600;
x = 100;
epoca = 5;
cont = 0;
for k1 = 1:1:epoca
    for  i = 1:1:t
        pacotes_images = pacotes{i};
        y_dados = Y_dados_pacotes{i};
        erro(cont + 1) = 0;
        for j=1:1:x
            A0= pacotes_images(:,j);
            Z1=W1*A0+B1;
            A1=sigmoide(Z1);
            Z2=W2*A1+B2;
            A2=sigmoide(Z2);
            Z3=W3*A2+B3;
            A3=sigmoide(Z3);
            y = y_dados(:,j); %vetor de rotulos
            k = A3-y; %Custo da rede neural
```

Fonte: elaborado pelo autor

Após concluir a fase inicial de processamento, o próximo passo é calcular o vetor gradiente, um componente essencial do algoritmo de *backpropagation*. A chave para esse cálculo é a derivada da função de ativação. Essa derivada é crucial porque determina como os pesos da rede neural são ajustados durante o treinamento. A implementação dessa função derivativa pode ser vista no pseudocódigo do Quadro 8:

Quadro 8 – Derivada da função sigmoide.

```
function [m] = sigmoide_linha(x)
m = exp(-x)./[1+exp(-x)+exp(-x)+exp(-2*x)];
end
```

Fonte: elaborado pelo autor

Com a derivada da função sigmoide, podemos fazer a implementação do algoritmo de *backpropagation*, descritos nas equações (16), (18), (28), (30), (32) e (33). No Quadro 9 apresentamos essa implementação:

A primeira parte do código é dedicada exclusivamente à etapa de treinamento da rede neural, utilizando um conjunto de dados que contém 60.000 imagens. Além disso, o conjunto de dados MNIST inclui um conjunto separado de 10.000 imagens destinadas ao teste da rede neural. Os arquivos 't10k-images.idx3-ubyte' e 't10k-labels.idx1-ubyte' armazenam, respectivamente, as imagens de teste e os rótulos correspondentes a cada uma delas.

Durante a fase de teste, o algoritmo processa as imagens e avalia o desempenho da rede neural, medindo o erro obtido com base no conjunto de teste, após o treinamento reali-

Quadro 9 – Implementação do algoritmo de *backpropagation*.

```
49              erro_por_imagem = sum(k.^2);
50              erro(cont + 1) = erro(cont+1) + erro_por_imagem;
51              SLINHA1=sigmoide_linha(Z1);
52              SLINHA2=sigmoide_linha(Z2);
53              SLINHA3=sigmoide_linha(Z3);
54              dc_db3=2.*SLINHA3.*k;
55              dc_dw3=2.*SLINHA3.*A2'.*k;
56              k_2 = SLINHA3.*W3.*2.*k;
57              dc_da2 =[sum(k_2(:,1));sum(k_2(:,2));sum(k_2(:,3));
                   sum(k_2(:,4));sum(k_2(:,5));sum(k_2(:,6));sum(
                   k_2(:,7));sum(k_2(:,8));sum(k_2(:,9));sum(k_2
                   (:,10));sum(k_2(:,11));sum(k_2(:,12));sum(k_2
                   (:,13));sum(k_2(:,14));sum(k_2(:,15));sum(k_2
                   (:,16))];
58              dc_db2 = SLINHA2.*dc_da2;
59              dc_dw2 = A2'.*SLINHA2.*dc_da2;
60              k3 = SLINHA2.*W2.*dc_da2;
61              dc_da1= [sum(k3(:,1));sum(k3(:,2));sum(k3(:,3));sum
                   (k3(:,4));sum(k3(:,5));sum(k3(:,6));sum(k3(:,7))
                   ;sum(k3(:,8));sum(k3(:,9));sum(k3(:,10));sum(k3
                   (:,11));sum(k3(:,12));sum(k3(:,13));sum(k3(:,14)
                   );sum(k3(:,15));sum(k3(:,16))];
62              dc_db1=SLINHA1.*dc_da1;
63              dc_dw1=A0'.*SLINHA1.*dc_da1;
64              alpha =0.2; %taxa de aprendizagem
65              W1 = W1 - alpha.*dc_dw1;
66              W2 = W2 - alpha.*dc_dw2;
67              W3 = W3 - alpha.*dc_dw3;
68              B1 = B1 - alpha.*dc_db1;
69              B2 = B2 - alpha.*dc_db2;
70              B3 = B3 - alpha*dc_db3;
71              end
72          erro(cont + 1) = erro(cont+1)/x;
73          cont = cont + 1;
74          plot(1:cont,erro)
75          pause(0.001)
76      end
77  end
```

Fonte: elaborado pelo autor

zado com as 60.000 imagens. Importante ressaltar que, nesta etapa, não ocorre o processo de *backpropagation*, pois o objetivo é avaliar a rede neural sem realizar ajustes adicionais nos seus parâmetros. Portanto, não há aprendizado contínuo da rede com base no conjunto de teste.

O pseudocódigo do Quadro 10, mostra a etapa de teste após o treinamento da rede neural:

O quadro completo da programação em Matlab encontra-se no anexo 6.1. Na seção seguinte, apresentaremos os resultados obtidos a partir da programação realizada no software Matlab.

Quadro 10 – Teste da eficiência da rede com 10.000 imagens.

```
images_test=loadMNISTImages('t10k-images.idx3-ubyte');
labels_test=loadMNISTLabels('t10k-labels.idx1-ubyte');
colunas_por_pacotes = 10000;
dados_teste = labels_vetor(labels_test);
tamanho_pacote_1 = repmat(colunas_por_pacotes,1, size(
   images_test,2)/colunas_por_pacotes);
Y_dados_tamanho_pacote_1 = repmat(colunas_por_pacotes,1, size(
   dados_teste,2)/colunas_por_pacotes);
pacotes_1 = mat2cell(images_test, size(images_test,1),
   tamanho_pacote_1);
Y_dados_pacotes_1 = mat2cell(dados_teste, size(dados_teste,1),
   tamanho_pacote_1);
s = 10000;
w = 1;
for a = 1:1:w
    pacotes_images = pacotes_1{a};
    y_dados = Y_dados_pacotes_1{a};
    erro_1 = 0;
for z = 1:1:s
        A0= pacotes_images(:,z);
        Z1=W1*A0+B1;
        A1=atv(Z1);
        Z2=W2*A1+B2;
        A2=atv(Z2);
        Z3=W3*A2+B3;
        A3=sigmoide(Z3);
        [n3,i3] = max(A3);
        y =  y_dados(:,z);
        [ny,iy] = max(y);
        if i3~=iy
            erro_1 = erro_1+1;
        end
end
        percent = erro_1/100
end
```

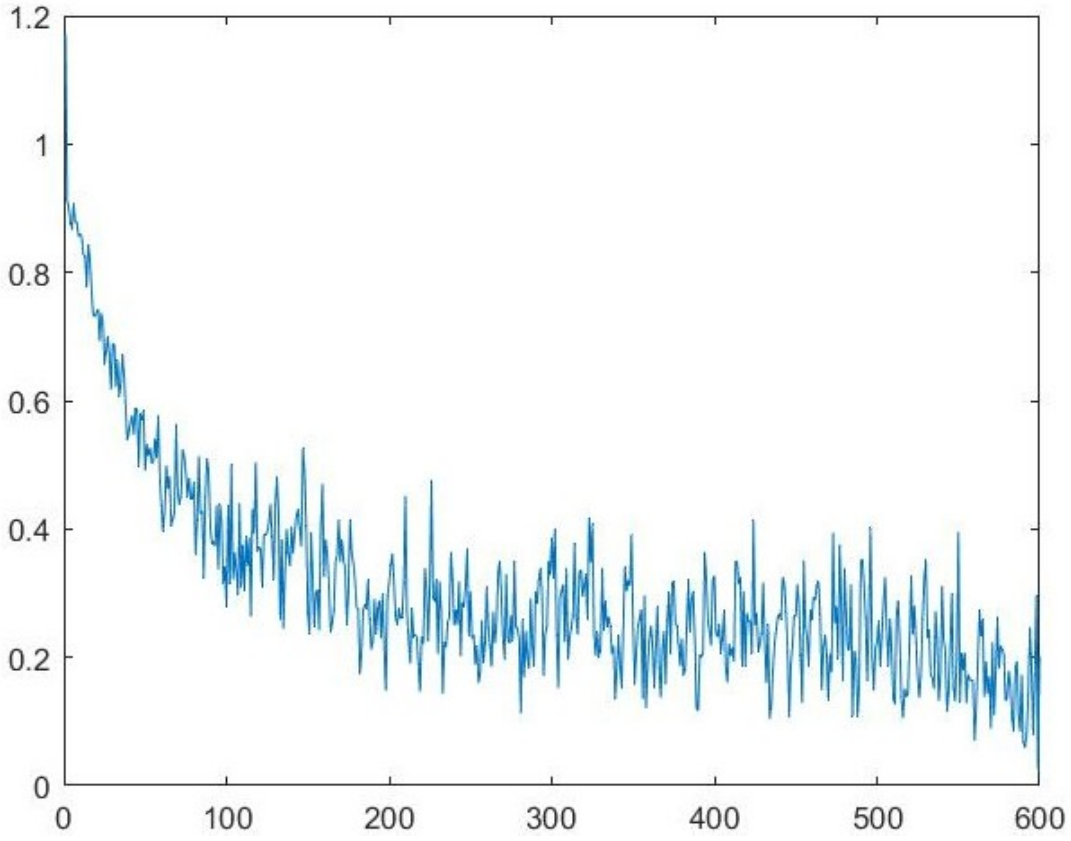
Fonte: elaborado pelo autor

3.6 Resultados obtidos no Matlab

O gráfico apresentado na Figura 16 mostra o treinamento da rede neural para uma época[4]. Temos o erro em função dos 600 pacotes de imagens. Durante o treinamento, após o processamento completo do conjunto de teste (época = 1), a rede neural apresentou uma taxa de erro de 13,36% em um total de 10.000 imagens.

[4] Cada época corresponde a uma iteração sobre todas as imagens do banco de dados de treinamento

Figura 16 – Gráfico do treinamento da rede neural.

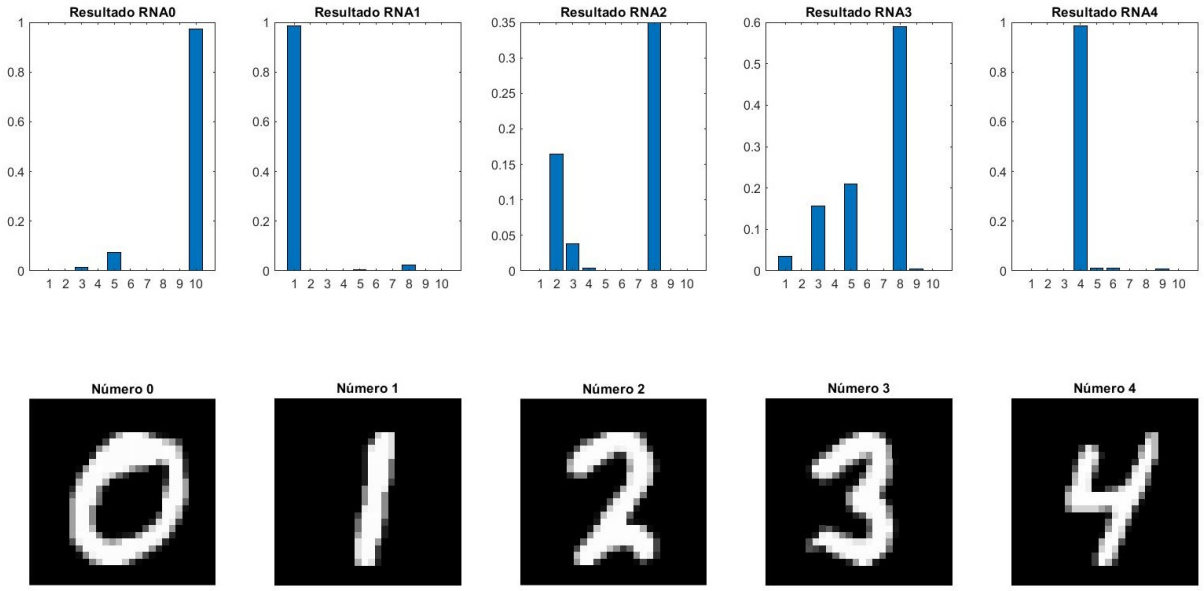


Fonte: elaborado pelo autor

A Figura 17 e 18 representam o gráfico de barras para as últimas imagens processadas no pacote de testes. Os resultados foram:

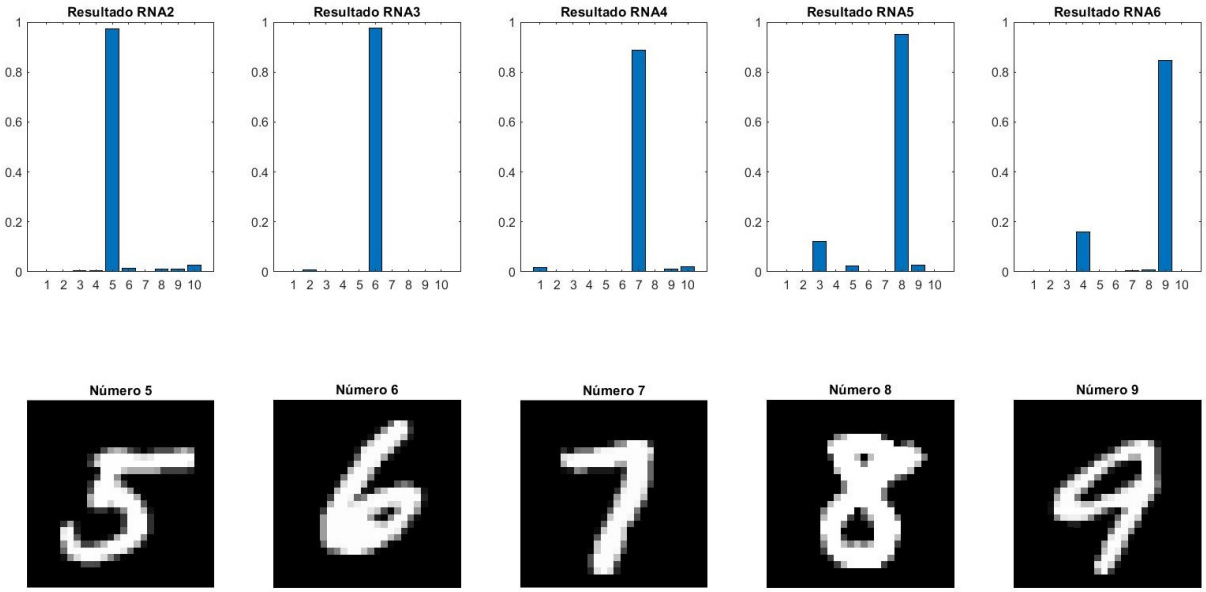
- 97,2% para o número 0;
- 98,6% para o número 1;
- 34,9% de chance do número 2 ser 8 e apenas 16,4% de ser um 2;
- 58,9% de chance do número 3 ser um 8 e apenas 15,6% de ser um 3;
- 98,4% para o número 4;
- 97,3% para o número 5;
- 97,6% para o número 6;
- 88,6% para o número 7;
- 95,1% para o número 8;
- 84,7,4% para o número 9;

Figura 17 – Resultado do processamento das últimas imagens do banco de testes.



Fonte: elaborado pelo autor

Figura 18 – Resultado do processamento das últimas imagens do banco de testes.



Fonte: elaborado pelo autor

Assim, concluímos os testes com nosso algoritmo de rede neural. A rede não continuou a evoluir, pois nosso objetivo era compreender o problema de aprendizado em redes neurais e explicar o processo, por meio de palestras, aos alunos do Ensino Médio.

4 DIVULGAÇÃO CIENTÍFICA: UMA PROPOSTA DE CONTRIBUIÇÃO PARA O ENSINO DE MATEMÁTICA.

Neste capítulo, discutiremos a experiência de desenvolver palestras na educação básica sobre a matemática envolvida na aprendizagem de máquinas e no desenvolvimento de inteligência artificial a partir de algoritmos de redes neurais artificiais (RNAs).

4.1 Palestras como objeto educacional de aprendizagem matemática

As palestras sobre aprendizado de máquinas e IA na educação básica desempenham um papel crucial ao preparar educadores e alunos para o mundo digital em evolução.

À medida que a IA torna-se mais presente em várias áreas do conhecimento e, consequentemente, na atuação de diversas profissões, é interessante que os estudantes compreendam seus conceitos básicos. Essas palestras capacitam os alunos a desenvolver habilidades relevantes para o mercado de trabalho, como análise de dados e automação, segundo (SILVA et al., 2023), a tecnologia revolucionou nossa vida cotidiana, e a educação não ficou para trás. Atualmente, é imperativo que os educadores adotem essa transformação e investiguem estratégias para integrar a tecnologia ao processo de ensino e aprendizagem. A tecnologia não é meramente uma ferramenta; ela atua como um catalisador no desenvolvimento de habilidades críticas nos alunos, incentivando-os a avaliar ideias e construir argumentos sólidos.

A IA faz parte do nosso cotidiano, desde assistentes virtuais até sistemas de recomendação em plataformas online. Por meio das palestras, podemos ajudar os alunos a se tornarem cidadãos digitais conscientes, capazes de avaliar criticamente as tecnologias e tomar decisões informadas. Além disso, discutir a IA envolve explorar questões éticas, como privacidade, viés algorítmico e transparência. As palestras incentivam reflexões sobre o uso responsável e ético dessas tecnologias.

A modelagem matemática e a aprendizagem de máquina estão intrinsecamente relacionadas, pois ambas envolvem conceitos matemáticos essenciais, como operações básicas entre matrizes e aplicações de funções. A "mágica" presente em uma rede neural nada mais é do que o conceito de função sendo calculada inúmeras vezes em uma aplicação. Mostrar aos alunos da educação básica que os conteúdos presentes no currículo são utilizados na vanguarda do desenvolvimento tecnológico é fundamental para a persistência do educando nos currículos de ciências exatas.

Neste contexto, a contribuição do estudo feito nesta dissertação para o desenvolvimento da educação básica foi na forma de palestras em 2 escolas no Município de Lucas do Rio Verde o Colégio Piaget e a Escola Estadual Militar Tiradentes SD PM Adriana Morais Ramos, no estado de Mato Grosso.

Essas palestras nas escolas de Lucas do Rio Verde, Mato Grosso, proporcionaram aos

alunos uma visão prática e concreta de como a matemática e a tecnologia se entrelaçam. Ao demonstrar como os conceitos matemáticos são aplicados na criação de modelos de aprendizado de máquina, os alunos puderam perceber que a Matemática não é apenas uma disciplina abstrata, mas sim uma ferramenta poderosa para resolver problemas do mundo real.

Além disso, o objetivo destas palestras foi incentivar o interesse dos alunos pelas ciências exatas, mostrando que esses conhecimentos não estão restritos aos livros didáticos, mas têm aplicações reais e impactam diretamente a sociedade. Ao compreenderem a relevância da matemática na IA, os alunos podem se sentir mais motivados a explorar carreiras relacionadas à tecnologia e à ciência.

Em suma, essas palestras não apenas enriqueceram o currículo escolar, mas também inspiraram os alunos a olhar para além das fórmulas e equações, reconhecendo o potencial transformador da matemática e da aprendizagem de máquina em suas vidas e no mundo ao seu redor.

4.2 O roteiro de palestra

A palestra foi pensada e desenvolvida para abranger certos tópicos da dissertação, que estão presentes na educação básica, tendo em vista as nossas hipóteses iniciais, que são: Como é construída uma inteligência artificial? Qual é a matemática presente no desenvolvimento das inteligências artificiais? Como uma inteligência artificial elabora uma resposta? E qual é o grau de confiança desta resposta?

Dessa forma, percorremos o roteiro abaixo, para produzir uma palestra em que o público alvo foram alunos do ensino médio.

1- Introdução: Abertura

- Pergunta impactante : "Com os avanços recentes e o crescimento explosivo das ferramentas de IA generativa, quais as suas expectativas para o impacto da IA?"
- Estatísticas Surpreendentes sobre IA
- Resultados surpreendentes (Chat Gpt, Dall-e, Boston Dynamics)
- O Poder do Aprendizado de Máquina
- Objetivo da palestra

2- Reconhecimento de números

- Imagem digital: imagem para matriz
- Rede neural: entrada e saída

- Neurônios, pesos e camadas
- Treinamento da rede neural
- Resultados

3- Recapitulação matemática

- **Matrizes** - Embora a Base Nacional Comum Curricular do Ensino Médio não tenha uma habilidade específica para matrizes e determinantes, dentro das diretrizes específicas para a matemática, esses conceitos estão intrinsecamente ligados a outras habilidades importantes para a formação do estudante. Ao trabalhar esses conteúdos, o professor contribui para o desenvolvimento de habilidades essenciais para o século XXI, como o pensamento crítico, a resolução de problemas e a capacidade de trabalhar com dados.
- **Representação binária** - a representação binária, embora não esteja explicitamente mencionada na BNCC, está intrinsecamente ligada a diversas habilidades, principalmente nas áreas de Matemática e Ciências da Natureza.
- **Funções**: A BNCC dedica um espaço significativo ao estudo de funções, reconhecendo a importância desse conceito para a compreensão de diversas áreas do conhecimento e para o desenvolvimento do pensamento matemático. As habilidades relacionadas a funções estão distribuídas ao longo do Ensino Fundamental e Médio, abrangendo diferentes aspectos como representação, análise e modelagem. Essas habilidades podem ser encontradas nos anos iniciais (EF01MA06, EF02MA09), nos anos finais (EF06MA09, EF07MA10, EF08MA13) e no Ensino Médio com as habilidades (EM13MAT302, EM13MAT503, EM13MAT507, EM13MAT508).
- **Cálculo de Mínimos (Otimização)**: Na BNCC, as habilidades relacionadas ao cálculo de mínimos estão predominantemente presentes no Ensino Médio, especialmente nas séries finais. Diversas habilidades envolvem a análise de mínimos de funções, destacando-se as habilidades EM13MAT503 e EM13MAT504.
- **Algoritmos e programação:** a BNCC aborda a programação e algoritmos principalmente por meio do pensamento computacional e da cultura digital. Exemplos disso incluem a habilidade EM13MAT405, que introduz conceitos básicos de programação e sua aplicação na resolução de problemas matemáticos, e a Competência 5, que trata da Cultura Digital, enfatizando a compreensão, utilização e criação de tecnologias digitais de forma crítica, significativa, reflexiva e ética.

4- Desafios e Futuro das Redes Neurais

- Mercado de trabalho/ oportunidades

- Perguntas e Discussão

4.3 Metodologia das Palestras

Para garantir que as palestras fossem educativas e engajantes, adotamos uma metodologia interativa que combinou explicações teóricas com demonstrações práticas e atividades colaborativas. A seguir, detalhamos a metodologia utilizada:

4.3.1 Preparação

Antes das palestras, realizamos reuniões estratégicas com a coordenação pedagógica do Colégio Piaget e da Escola Estadual Militar Tiradentes SD PM Adriana Morais Ramos em 04 de Abril de 2024. O objetivo desses encontros foi sincronizar as metas da palestra com as necessidades específicas e o nível de conhecimento dos estudantes. A Tabela 1 mostra o cronograma de encontros com a coordenação e alinhamentos com o orientador antes da realização das palestras:

Tabela 1 – Cronograma de atividades

Datas	Reuniões
11/04/2024	Conversa com a coordenação e direção de cada escola para realização da palestra
18/04/2024	Encaminhamentos com os professores que cederam as aulas
25/04/2024	Reunião com o orientador para últimos encaminhamentos sobre as palestras
03/05/2024	Palestra na Escola Estadual Militar Tiradentes SD PM Adriana Morais Ramos
10/05/2024	Palestra no Colégio Piaget

Na Escola Estadual Militar Tiradentes SD PM Adriana Morais Ramos, a palestra foi conduzida no dia 03 de maio de 2024, contando com a presença de 60 alunos do 3° ano do ensino médio. Já no Colégio Piaget, a apresentação ocorreu no dia 10 de maio de 2024, engajando um diversificado grupo de 140 estudantes dos 1°, 2° e 3° anos do ensino médio.

Durante as palestras, coletamos *feedback* por meio do questionário no Google Formulários e utilizamos a plataforma Mentimeter para criar nuvens de palavras sobre a percepção dos alunos em relação ao tema. A nuvem de palavras é uma ferramenta visual que representa as palavras mais relevantes em um determinado texto ou conjunto de dados. Ela é frequentemente usada para destacar termos-chave e padrões em análises de conteúdo. Em uma Nuvem de Palavras, as palavras mais frequentes aparecerão com maior destaque na imagem.

A nuvem de palavras é útil em várias situações, como análise de texto, interpretação de dados, currículos, didática e até mesmo em projetos de *brainstorming*.A análise desses resul-

tados nos ajudou a avaliar o sucesso das palestras e a identificar áreas para aprimoramento em futuras iniciativas educacionais, como podemos observar na Figura 20.

4.3.2 Execução

1- Introdução: Abertura

- **Pergunta impactante:** Iniciamos a palestra com os alunos criando uma nuvem de palavras sobre: "Com os avanços recentes e o crescimento explosivo das ferramentas de IA generativa, quais as suas expectativas para o impacto da IA? e logo após passamos o com vídeo sobre o desenvolvimento de robôs autônomos da empresa Boston Dynamics[1]

Figura 19 – Nuvem de palavras referente às respostas obtidas para a pergunta:"Com os avanços recentes e o crescimento explosivo das ferramentas de IA generativa, quais as suas expectativas para o impacto da IA?

Fonte: elaborado pelos alunos, na plataforma Mentimeter, no Colégio Piaget

- **Estatísticas Surpreendentes sobre IA:** Apresentamos dados sobre o crescimento e impacto da IA em diversas áreas, destacando seu uso cotidiano.
- **Resultados surpreendentes:** Demonstramos tecnologias como ChatGPT, DALL-E e os robôs da Boston Dynamics, para exemplificar os avanços impressionantes da IA.
- **O Poder do Aprendizado de Máquina:** Explicamos brevemente como o aprendizado de máquina permite que sistemas aprendam e melhorem com a experiência.

[1] $https://www.youtube.com/watch?v =_E ZQx87DyzM$

- **Objetivo da palestra:** Delineamos os objetivos da palestra, que incluíam proporcionar uma compreensão básica da IA e mostrar a aplicação prática da matemática.

2- Reconhecimento de números

- **Imagem digital: imagem para matriz:** Mostramos como uma imagem digital é convertida em uma matriz de números.
- **Rede neural: entrada e saída:** Explicamos o funcionamento básico de uma rede neural, destacando os conceitos de entrada (input) e saída (output).
- **Neurônios, pesos e camadas:** Descrevemos a estrutura interna das redes neurais, incluindo neurônios, pesos e camadas.
- **Treinamento da rede neural:** Demonstramos o processo de treinamento de uma rede neural usando o problema de reconhecimento de imagens feito neste trabalho.
- **Resultados:** Apresentamos os resultados de redes neurais treinadas, mostrando a precisão e as aplicações práticas.

3- Recapitulação matemática

- **Matrizes:** Revisamos o conceito de matrizes e sua importância no processamento de imagens e em redes neurais.
- **Representação binária:** Explicamos como os dados são representados binariamente e utilizados nas computações.
- **Funções:** Abordamos o papel das funções matemáticas no aprendizado de máquina, especialmente nas operações de ativação em redes neurais.
- **Cálculo de mínimos (otimização):** Introduzimos técnicas de otimização, como gradiente descendente, usadas para minimizar erros em redes neurais.
- **Programação/algoritmos:** Discutimos a importância da programação e dos algoritmos no desenvolvimento e aplicação de modelos de IA.

4- Desafios e Futuro das Redes Neurais

- **Mercado de trabalho/oportunidades:** Exploramos as oportunidades de carreira na área de IA e como o mercado de trabalho está se transformando.
- **Perguntas e Discussão:** Abrimos espaço para perguntas dos alunos e uma discussão interativa sobre os tópicos abordados, incentivando a participação ativa.

4.4 Impacto das Palestras nas Escolas

As palestras realizadas nas escolas de Lucas do Rio Verde não só abordaram os conceitos teóricos de Inteligência Artificial e da Matemática aplicada, mas também proporcionaram atividades práticas que envolvem os alunos diretamente no processo de aprendizado. Isso permitiu que os alunos vissem, em primeira mão, como os princípios matemáticos que eles estudam na sala de aula são aplicados em tecnologias modernas.

A recepção dos alunos foi extremamente positiva. Muitos demonstraram um interesse renovado pela matemática e pelas ciências exatas, reconhecendo a relevância dessas disciplinas para a compreensão e desenvolvimento de tecnologias avançadas. Além disso, os professores relataram um aumento no engajamento dos alunos nas aulas subsequentes, o que sugere que as palestras tiveram um impacto duradouro na motivação dos alunos.

Figura 20 – "Escreva uma palavra que expressa a sua avaliação sobre o tema e a relevância da palestra".

Fonte: elaborado pelos alunos no final da palestra na EE Militar Tiradentes SD PM Adriana Morais Ramos

A análise da nuvem de palavras gerada a partir das respostas dos estudantes revela que o tema desperta interesse significativo entre os discentes. A maioria dos estudantes considera o tema relevante e importante, evidenciando sua pertinência no contexto educacional.

Na próxima seção, faremos à análise das respostas fornecidas pelos alunos em relação a tópicos e perguntas específicas sobre a palestra que assistiram.

4.5 Questionários

A escala de Likert é amplamente reconhecida como a ferramenta padrão em pesquisas de opinião e atitudes. Desenvolvida em 1932 pelo renomado professor e pesquisador Rensis Likert, essa escala é essencial para quantificar percepções e reações, segundo (BERMUDES et al., 2016, p.14). Com base nessa metodologia, adotamos a escala de Likert para avaliar o impacto de nossa palestra no público-alvo.

Para capturar esses dados, desenvolvemos um questionário detalhado utilizando o Google Formulários, que foi prontamente disponibilizado aos alunos ao término da palestra. Esse instrumento nos permitiu coletar *feedback* valioso e mensurar a eficácia da apresentação de maneira estruturada e sistemática.

O formulário foi elaborado para captar a eficácia da palestra em termos de aumento da compreensão e estímulo ao interesse dos alunos pela matemática, especialmente no contexto das redes neurais e do desenvolvimento tecnológico. As perguntas foram estruturadas para abranger uma gama de respostas, desde a ausência de impacto até um aumento significativo na compreensão e no encorajamento para aprofundar-se no assunto.

As questões propostas visavam avaliar, mesmo que subjetivamente, os seguintes pontos:

- **Aumento da Compreensão**: Se a palestra contribuiu para uma melhor percepção de como a matemática é fundamental no desenvolvimento de tecnologias avançadas.
- **Incentivo ao Aprendizado**: Se os alunos se sentiram motivados a explorar mais sobre a matemática por trás das redes neurais após a palestra.
- **Esclarecimento de Conceitos**: Se a apresentação de exemplos práticos foi efetiva em tornar os conceitos mais claros.
- **Clareza da Apresentação**: A qualidade da exposição dos temas e a facilidade com que foram comunicados.
- **Relevância do Conteúdo**: Se os temas abordados foram pertinentes e interessantes para os alunos.
- **Comunicação do Palestrante**: A eficácia com que o palestrante transmitiu as informações.
- **Engajamento**: O quão cativante e envolvente foi a palestra.
- **Duração Apropriada**: Se o tempo dedicado à palestra foi adequado.
- **Recomendação**: A disposição dos alunos em recomendar a palestra a outros.
- **Utilidade**: Como a palestra se alinhou com os estudos e interesses pessoais dos alunos.

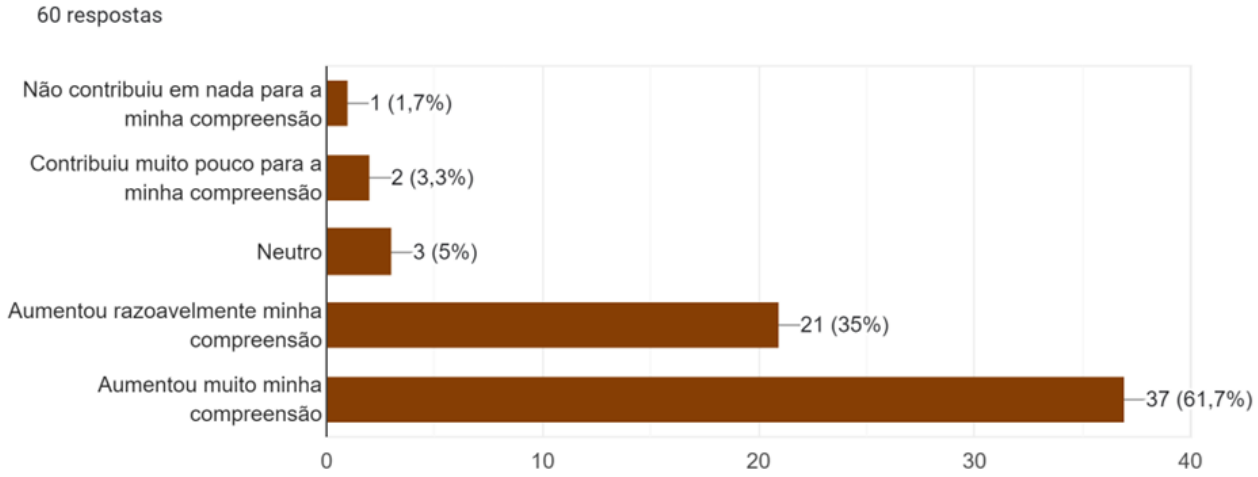
Essas métricas nos permitiram não apenas medir o sucesso da palestra, mas também identificar áreas para melhorias futuras. A análise das respostas forneceu *insights* valiosos sobre a percepção dos alunos e ajudou a moldar iniciativas educacionais subsequentes para maximizar o engajamento e o aprendizado.

4.6 Resultados do questionário

No contexto do evento, foi enfatizado que a participação no questionário era inteiramente voluntária. As questões foram cuidadosamente projetadas para serem não-obrigatórias, permitindo aos participantes a plena autonomia de escolha. Assim, cada indivíduo teve a liberdade de decidir se contribuiria com suas respostas ao questionário. Além disso, dentro do próprio formulário, nenhuma das perguntas foi estabelecida como compulsória, proporcionando aos participantes a flexibilidade de selecionar as questões que preferiam responder. Essa abordagem assegurou que a coleta de dados refletisse genuinamente o interesse e o engajamento dos participantes, garantindo a integridade e a relevância das informações coletadas.

A primeira questão do questionário focou no aumento da compreensão dos estudantes sobre os conceitos matemáticos aplicados no desenvolvimento de tecnologias de Inteligência Artificial. Os resultados obtidos nas duas escolas são ilustrados no gráfico subsequente, que destaca a percepção dos alunos quanto ao aprofundamento de seu entendimento nessa área.

Figura 21 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: a palestra aumentou sua compreensão de que a Matemática é utilizada no desenvolvimento de tecnologias?



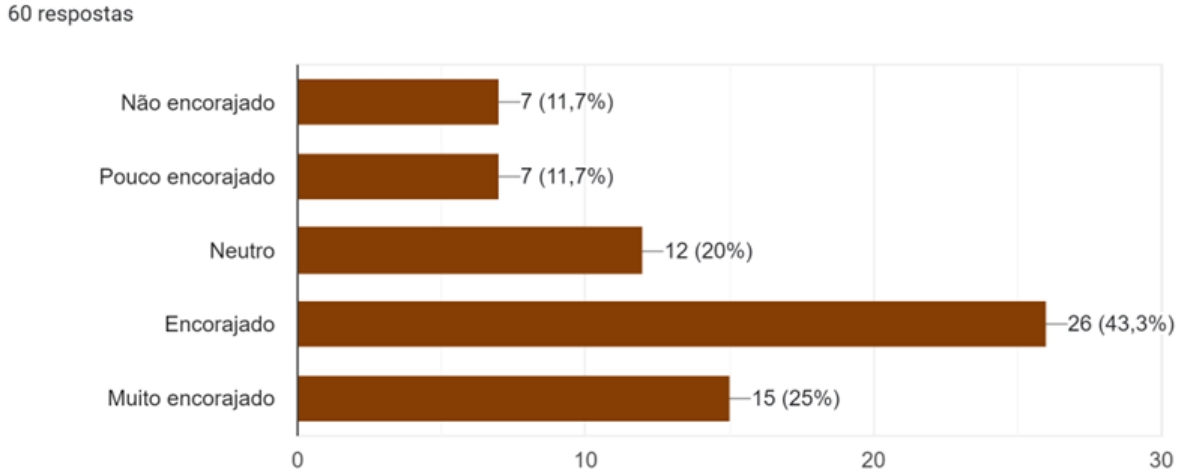
Fonte: elaborado pelo autor

Os resultados indicam uma compreensão significativa por parte dos alunos sobre o papel vital da matemática no desenvolvimento de redes neurais. A relevância dos estudos matemáticos nesse campo é, sem dúvida, fundamental e bem reconhecida e foi amplamente enfatizado

na palestra.

A segunda pergunta abordou a motivação dos alunos para explorar mais a matemática envolvida no desenvolvimento dessas tecnologias. Considerando que estamos lidando com uma geração que cresceu com acesso amplo à cultura digital e iteração online, a maioria dos alunos que participaram da palestra sentiu-se encorajada a aprofundar seus conhecimentos nessa temática. O gráfico abaixo esboça esses resultados:

Figura 22 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: você se sente encorajado a explorar mais a matemática por trás das redes neurais após assistir à palestra?



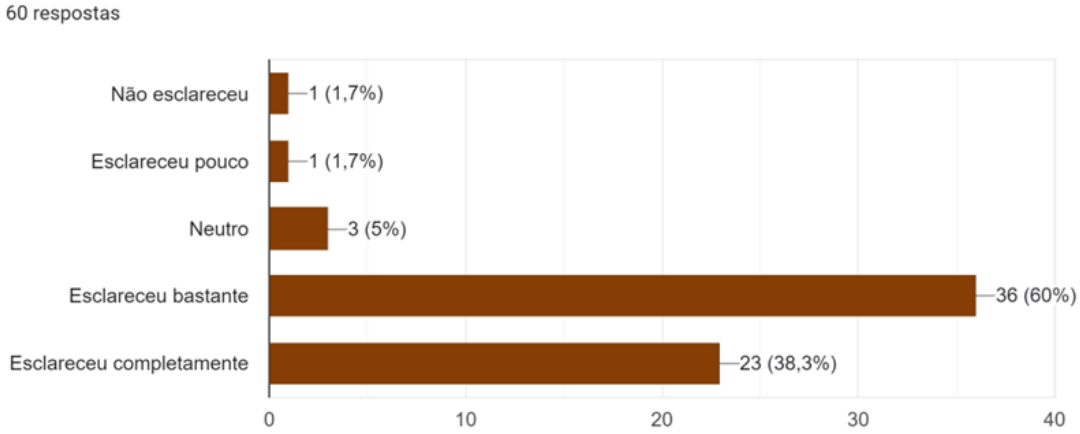
Fonte: elaborado pelo autor

A palestra concentrou-se em apresentar a rede neural desenvolvida nesta dissertação, que é o algoritmo de reconhecimento de dígitos manuscritos para exemplificar como é o processo de construção de uma inteligência artificial. Assim, os conceitos relacionados aos dados de entrada do pacote MNIST, o processamento da rede e os dados gerados na saída da rede foram abordados durante toda a apresentação e foram verificados na Figura 24 para garantir que os alunos compreenderam os conceitos apresentados:

- Compreensão Total: 60% dos participantes indicaram que a palestra “Esclareceu completamente” os conceitos, demonstrando uma alta taxa de entendimento.
- Compreensão Significativa: 38,3% sentiram que a palestra “Esclareceu bastante”, sugerindo que a maioria dos alunos conseguiu assimilar bem as informações.
- Neutro: Uma minoria de 5% se posicionou como “Neutro”, o que pode indicar a necessidade de revisões ou esclarecimentos adicionais.
- Pouco ou Nenhum Esclarecimento: Apenas 3,4% dos alunos sentiram que a palestra “Esclareceu pouco” ou “Não esclareceu”, apontando para uma eficácia geral muito boa da apresentação.

Essa análise reforça a efetividade da palestra em transmitir conhecimento sobre um tópico complexo, com uma clara maioria dos alunos sentindo-se esclarecidos sobre o assunto. Isso sugere que a metodologia e os recursos utilizados foram adequados para o ensino do conteúdo proposto.

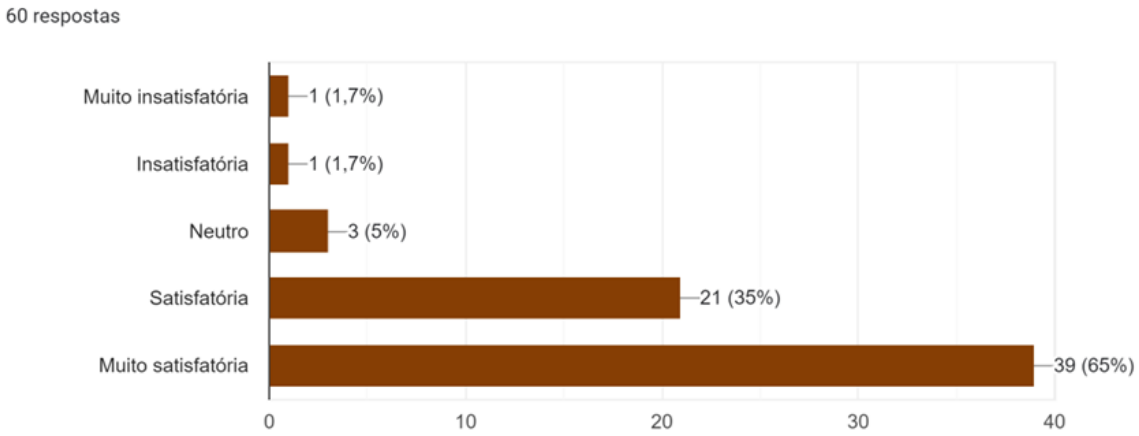
Figura 23 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: a apresentação de conceitos práticos ajudou a esclarecer os conceitos apresentados?



Fonte: elaborado pelo autor

A quarta pergunta da avaliação focou na qualidade da apresentação do tema e na capacidade do palestrante em comunicar a mensagem proposta. Esta questão é crucial para entender como o público percebeu a eficácia do orador em engajar a audiência e transmitir claramente os pontos-chave do assunto abordado.

Figura 24 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: como você classifica a clareza da apresentação?

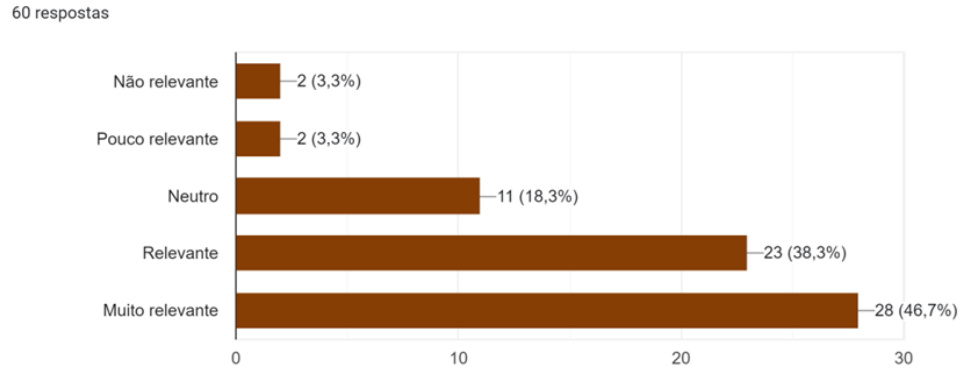


Fonte: elaborado pelo autor

A quinta pergunta buscou avaliar a relevância da inteligência artificial para um grupo de alunos do ensino médio, que, indubitavelmente, encontrarão um mercado de trabalho em

evolução devido ao impacto desta tecnologia. Essa questão é essencial para compreender o interesse e a percepção dos estudantes sobre como a inteligência artificial poderá influenciar suas futuras carreiras profissionais:

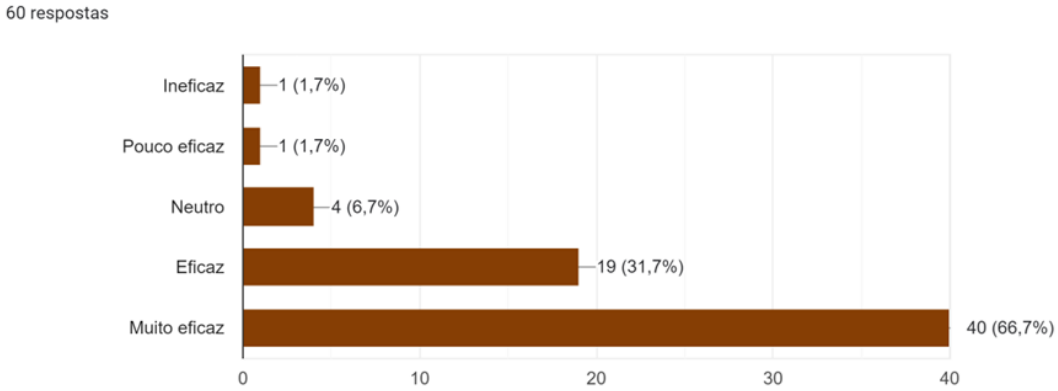
Figura 25 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: a palestra abordou temas relevantes e de seu interesse?



Fonte: elaborado pelo autor

A sexta questão do questionário, assim como a quarta, foi elaborada com o objetivo de avaliar a organização e a clareza da apresentação. Especificamente, buscou-se entender o quão eficiente foi o palestrante na condução do evento, considerando aspectos como fluidez, coerência e engajamento do público. A imagem abaixo representando os resultados, com a maioria das respostas indicando "Muito eficaz" (66%) e "Eficaz" (31,67%). Isso sugere que os participantes consideraram em sua maioria que o palestrante comunicou-se eficientemente.

Figura 26 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: a comunicação do palestrante foi eficaz?

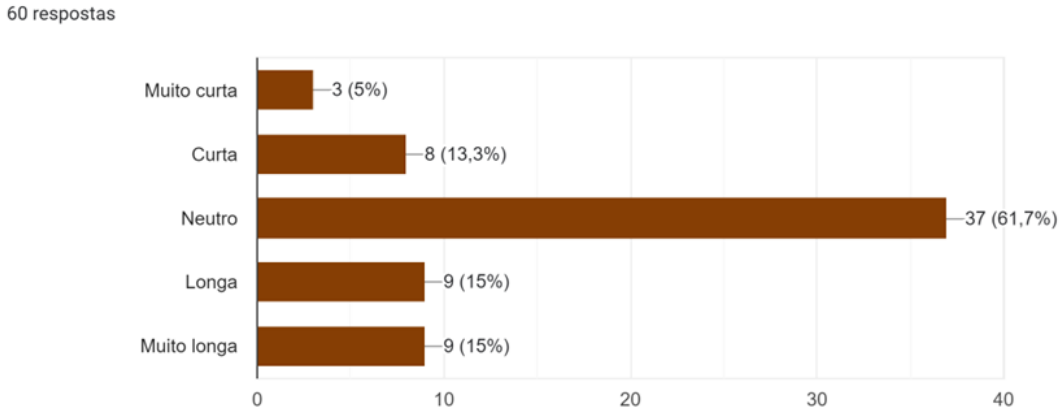


Fonte: elaborado pelo autor

A análise da pergunta sétima questão, que abordou a adequação do tempo de duração da palestra, revelou uma divisão de opiniões entre os participantes. Enquanto a maioria, com 61,7%, considerou a duração como neutra, indicando satisfação com o tempo estabelecido, 18,3% dos alunos relataram que a palestra foi curta ou muito curta, sugerindo um desejo de

prolongamento. Por outro lado, 30% dos respondentes acharam a palestra longa ou muito longa, refletindo uma preferência por sessões mais breves. Essas informações são valiosas para ajustar a duração das palestras futuras, equilibrando as expectativas e preferências do público-alvo.

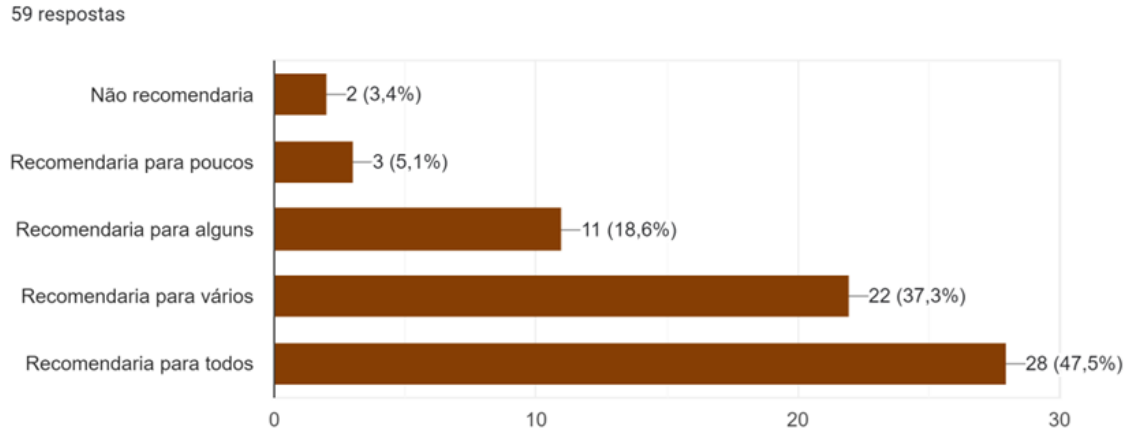
Figura 27 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: a duração da palestra foi apropriada?



Fonte: elaborado pelo autor

A análise do gráfico da oitava questão questiona se os participantes da palestra recomendariam a mesma para outros alunos indica uma recepção positiva do evento. Com 47,5% dos respondentes afirmando que "Recomendaria para todos" e 37,3% que "Recomendaria para vários", fica evidente que a maioria dos participantes considerou a palestra valiosa e digna de ser compartilhada. Apenas uma minoria de 8,5% optou pelas categorias "Não recomendaria" ou "Recomendaria para poucos", o que sugere que o conteúdo e a entrega da palestra foram bem-recebidos e tiveram um impacto positivo na audiência. Esses resultados são encorajadores e destacam a eficácia da palestra em engajar e informar os alunos.

Figura 28 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: você recomendaria esta palestra para outros alunos?



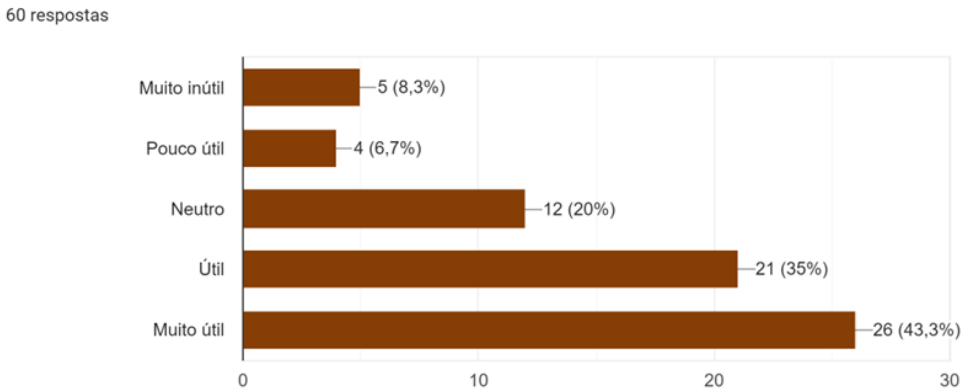
Fonte: elaborado pelo autor

O gráfico sobre a última questão do nosso questionário, indaga os alunos do ensino médio sobre o "Quão útil foi a palestra no contexto dos seus estudos e interesses pessoais?" revelando a percepção dos alunos deste ciclo educacional sobre a utilidade de uma palestra focada em inteligência artificial e a matemática envolvida no desenvolvimento de redes neurais. A análise detalhada é a seguinte:

- Muito útil: 26 respostas (43,3%) indicam que quase metade dos alunos considerou a palestra extremamente benéfica para seus estudos e interesses.
- Útil: 21 respostas (35%) mostram que muitos alunos encontraram valor na palestra, reforçando sua relevância educacional.
- Neutro: 12 respostas (20%) sugerem que alguns alunos estavam indecisos quanto à utilidade da palestra.
- Pouco útil e Muito inútil: 9 respostas (15%) combinadas indicam que uma minoria dos alunos não percebeu um grande benefício da palestra.

Esses dados sugerem um forte interesse ou valor percebido nos tópicos de IA e Matemática entre os alunos, com uma maioria expressiva reconhecendo a importância da palestra para seu desenvolvimento acadêmico e pessoal. A palestra parece ter sido bem recebida e considerada uma adição valiosa à educação dos alunos.

Figura 29 – Gráfico de barras elaborado a partir das respostas dos alunos para a pergunta: quão útil foi a palestra no contexto dos seus estudos e interesses pessoais?



Fonte: elaborado pelo autor

5 CONCLUSÕES

As redes neurais artificiais são verdadeiramente fascinantes. Elas têm o potencial de revolucionar diversos setores da sociedade, desde a medicina até as finanças e a automação. Embora ainda haja muito a ser descoberto e desenvolvido, já podemos vislumbrar os benefícios que essas redes estão trazendo.

A tecnologia abre novas e inesperadas inovações em nossas vidas, bem como cria oportunidades para geração de riquezas, conhecimento e aumento da qualidade de vida. Fazer com que estudantes tornem-se conscientes dessa realidade pode estimulá-los a tornarem-se eles mesmos participantes desse incrível processo de avanço.

As redes neurais artificiais são apenas um exemplo de como a matemática impulsiona a inovação tecnológica. Ao dominar conceitos básicos do ensino médio, você abre um leque de oportunidades para explorar o fascinante mundo da inteligência artificial e contribuir para o futuro da tecnologia. A matemática não é apenas um conjunto de fórmulas e teoremas, mas sim uma ferramenta poderosa para desvendar os segredos do mundo e criar novas tecnologias.

O crescimento exponencial no uso de algoritmos de redes neurais, impulsionado pelas inteligências artificiais generativas, pareceu, por muito tempo, algo reservado às grandes empresas e corporações. No entanto, esses algoritmos são baseados em programações acessíveis e totalmente viáveis de serem implementadas. Eles têm o potencial de resolver problemas específicos na sociedade, tornando o desenvolvimento dessas tecnologias algo tangível e promissor.

A realização de palestras sobre redes neurais na educação básica tem um propósito claro e empolgante: revelar aos estudantes do ensino médio que a matemática é a força propulsora por trás do desenvolvimento da Inteligência Artificial (IA). Essas palestras visam demonstrar que a IA não é apenas uma carreira do futuro, mas uma área emergente e acessível, ao alcance dos jovens de hoje.

Ao entenderem que o desenvolvimento dessas tecnologias avançadas está intimamente ligado ao conhecimento matemático, os estudantes podem ver a matemática sob uma nova luz, não como um conjunto de equações abstratas, mas como uma ferramenta poderosa para inovação e progresso.

REFERÊNCIAS

ANDRYCHOWICZ, M.; BAKER, B.; CHOCIEJ, M.; JOZEFOWICZ, R.; MCGREW, B.; PACHOCKI, J.; PETRON, A.; PLAPPERT, M.; POWELL, G.; RAY, A. et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, arXiv, 2018.

BENJAMIN. *Sunspring*. 2016. Curta-metragem escrito e produzido por um sistema de IA. Desenvolvido por pesquisadores do Instituto de Tecnologia de Massachusetts.

BERMUDES, W. L.; SANTANA, B. T.; BRAGA, J. H. O.; SOUZA, P. H. Tipos de escalas utilizadas em pesquisas e suas aplicações. *VÉRTICES*, Campos dos Goytacazes/RJ, 2016. Disponível em: <https://editoraessentia.iff.edu.br/index.php/vertices/article/download/1809-2667.v18n216-01/5242>.

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDEMIR, T. B. Redes neurais artificiais: teoria e aplicações. *Rio de Janeiro: LTC*, 2000.

BROWN, N.; SANDHOLM, T. Superhuman ai for multiplayer poker. *Science*, American Association for the Advancement of Science, v. 365, n. 6456, p. 885–890, 2019.

GRADIENT. Gradient descent. In: ____. *Wikipédia: a enciclopédia livre*. Wikimedia, 2024. Disponível em: <https://en.wikipedia.org/wiki/Gradient_descent>. Acesso em: 08 maio 2024.

HAYKIN, S. *Redes Neurais: princípios e práticas*. [S.l.]: Porto Alegre: Bookman, 2007.

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.

NETO, A. B.; BONINI, C. d. S. B. Redes neurais artificiais: Apresentação e utilização do algoritmo perceptron em biossistemas. *BioEng, Tupã*, 2010.

NIELSEN M., A. *Neural Networks and Deep Learning,*. Determination Press, 2015. Acesso em 13 de maio, 2024. Disponível em: <http://neuralnetworksanddeeplearning.com/>.

SANDERSON, G. *3blue1brown*. 2016. Acesso em 13 de maio, 2024. Disponível em: <https://www.3blue1brown.com/>.

SHUBHIRAJ, M.; FRANKS, L.; LU, P. *O Banco de dados MNIST de dígitos manuscritos*. 2023. Acesso em 13 de maio, 2024. Disponível em: <https://tinyurl.com/4bc89kut/>.

SILVA, J. E. M. da; CARVALHO, E. T. de; FERREIRA, S. C.; ESCOBAR, D. M. Redes neurais e inteligÊncias artificiais, seus impactos na sociedade e no ensino escolar: Uma revisÃo da literatura. *Revistaft*, 2023. Acesso em 18 de maio, 2024. Disponível em: <https://revistaft.com.br/redes-neurais-e-inteligencias-artificiais-seus-impactos-nasociedade-e-no-ensino-escolar-uma-revisao-da-l
>.

SILVER, D.; HUANG, A.; MADDISON, C. J.; GUEZ, A.; SIFRE, L.; DRIESSCHE, G. V. D.; SCHRITTWIESER, J.; ANTONOGLOU, I.; PANNEERSHELVAM, V.; LANCTOT, M. et al. Mastering the game of go with deep neural networks and tree search. *Nature*, Nature Publishing Group, v. 529, n. 7587, p. 484–489, 2016.

STEWART, J. *Cálculo, volume 2*. 7. ed. [S.l.]: São Paulo : Cengage Learning, 2013.

WARWICK, K.; SHAH, H. Can machines think? a report on turing test experiments at the royal society. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 28, n. 6, p. 989–1007, 2016.

6 APÊNDICE A

6.1 A programação completa em MATLAB

Listing 6.1 – Código completo da iteração principal.

```
clear all
close all

images=loadMNISTImages('train-images.idx3-ubyte'); % vetor 784
   x 60.000

labels_dados = loadMNISTLabels('train-labels.idx1-ubyte');

Y_dados=labels_vetor(labels_dados);


L0=784;
L1=16;
L2=16;
L3=10;

W1=randn(L1,L0);
B1=randn(L1,1);
W2=randn(L2,L1);
B2=randn(L2,1);
W3=randn(L3,L2);
B3=randn(L3,1);

% n mero de coluna por pacote
colunas_por_pacotes = 100;
%Divis o das imagens em pacotes de 100
tamanho_pacote = repmat(colunas_por_pacotes,1, size(images,2)/
   colunas_por_pacotes);
%divis o dos r tulos em pacotes de 100
Y_dados_tamanho_pacote = repmat(colunas_por_pacotes,1, size(
   Y_dados,2)/colunas_por_pacotes);

%pacotes com o comando mat2cell

```

```
32 pacotes = mat2cell(images, size(images,1), tamanho_pacote);
33 %divis o dos r tulos
34 Y_dados_pacotes = mat2cell(Y_dados, size(Y_dados,1),
      tamanho_pacote);
35
36 t = 600;
37 x = 100;
38 figure(1);
39 cont = 0;
40
41 epoca = 5;
42
43 for k1 = 1:1:epoca
44 for  i = 1:1:t
45
46     pacotes_images = pacotes{i};
47     y_dados = Y_dados_pacotes{i};
48
49     erro(cont + 1) = 0;
50
51     for j=1:1:x
52
53
54         A0= pacotes_images(:,j);
55
56
57         Z1=W1*A0+B1;
58
59         A1=atv(Z1);
60
61         Z2=W2*A1+B2;
62
63         A2=atv(Z2);
64
65         Z3=W3*A2+B3;
66
67         A3=sigmoide(Z3);
68
69         %custo da rede
```

```
70
71          SLINHA1=sigmoide_linha(Z1);
72
73          SLINHA2=sigmoide_linha(Z2);
74
75          SLINHA3=sigmoide_linha(Z3);
76
77          y = y_dados(:,j);
78
79          k = A3-y;
80
81          erro_por_imagem = sum(k.^2);
82          erro(cont + 1) = erro(cont+1) + erro_por_imagem;
83
84
85
86          dc_db3=2.*SLINHA3.*k;
87
88          dc_dw3=2.*SLINHA3.*A2'.*k;
89
90
91
92          k_2 = SLINHA3.*W3.*2.*k;
93
94          dc_da2 =[sum(k_2(:,1));sum(k_2(:,2));sum(k_2(:,3));sum(
               k_2(:,4));sum(k_2(:,5));sum(k_2(:,6));sum(k_2(:,7));
               sum(k_2(:,8));sum(k_2(:,9));sum(k_2(:,10));sum(k_2
               (:,11));sum(k_2(:,12));sum(k_2(:,13));sum(k_2(:,14))
               ;sum(k_2(:,15));sum(k_2(:,16))];
95
96          dc_db2 = SLINHA2.*dc_da2;
97
98          dc_dw2 = A2'.*SLINHA2.*dc_da2;
99          %derivada 1 camada
100
101         k3 = SLINHA2.*W2.*dc_da2;
102
103         dc_da1= [sum(k3(:,1));sum(k3(:,2));sum(k3(:,3));sum(k3
               (:,4));sum(k3(:,5));sum(k3(:,6));sum(k3(:,7));sum(k3
```

```
            (:,8));sum(k3(:,9));sum(k3(:,10));sum(k3(:,11));sum(
            k3(:,12));sum(k3(:,13));sum(k3(:,14));sum(k3(:,15));
            sum(k3(:,16))];
104
105         dc_db1=SLINHA1.*dc_da1;
106
107         dc_dw1=A0'.*SLINHA1.*dc_da1;
108         alpha =0.2;
109         W1 = W1 - alpha.*dc_dw1;
110         W2 = W2 - alpha.*dc_dw2;
111         W3 = W3 - alpha.*dc_dw3;
112         B1 = B1 - alpha.*dc_db1;
113         B2 = B2 - alpha.*dc_db2;
114         B3 = B3 - alpha*dc_db3;
115
116
117     end
118     erro(cont + 1) = erro(cont+1)/x;
119     cont = cont + 1;
120     plot(1:cont,erro)
121     pause(0.001)
122
123 end
124
125
126 end
127
128 % pacote de teste
129 images_test=loadMNISTImages('t10k-images.idx3-ubyte'); % 10.000
        imagens para testes
130 labels_test=loadMNISTLabels('t10k-labels.idx1-ubyte'); % 10.000
        resultados
131 colunas_por_pacotes = 10000;
132 dados_teste = labels_vetor(labels_test);
133
134
135 tamanho_pacote_1 = repmat(colunas_por_pacotes,1, size(
       images_test,2)/colunas_por_pacotes);
136 %divis o dos r tulos em pacotes de 100
```

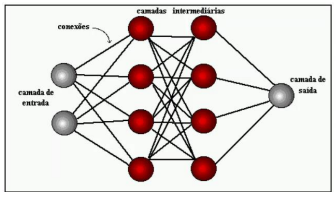
```
137 Y_dados_tamanho_pacote_1 = repmat(colunas_por_pacotes,1, size(
       dados_teste,2)/colunas_por_pacotes);
138
139 %pacotes com o comando mat2cell
140
141 pacotes_1 = mat2cell(images_test, size(images_test,1),
       tamanho_pacote_1);
142 %divis o dos r tulos
143 Y_dados_pacotes_1 = mat2cell(dados_teste, size(dados_teste,1),
       tamanho_pacote_1);
144
145
146
147 s = 10000;
148 w = 1;
149
150
151 for a = 1:1:w
152     pacotes_images = pacotes_1{a};
153     y_dados = Y_dados_pacotes_1{a};
154
155     erro_1 = 0;
156 for z = 1:1:s
157         A0= pacotes_images(:,z);
158
159
160
161         Z1=W1*A0+B1;
162
163         A1=atv(Z1);
164
165         Z2=W2*A1+B2;
166
167         A2=atv(Z2);
168
169         Z3=W3*A2+B3;
170
171         A3=sigmoide(Z3);
172
```

```
173         [n3,i3] = max(A3);
174
175         y =  y_dados(:,z);
176
177         [ny,iy] = max(y);
178
179         if i3~=iy
180             erro_1 = erro_1+1;
181         end
182
183
184
185
186
187 end
188
189         percent = erro_1/100
190
191 end
```

6.2 Slides da palestra

UNIVERSIDADE DO ESTADO DE MATO GROSSO
CÂMPUS UNIVERSITÁRIO DE SINOP
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS

PROFMAT



A MATEMÁTICA NAS REDES NEURAIS

Mestrando: Douglas Sales Pauli
Orientador: Dr. Raul de Abreu Assis

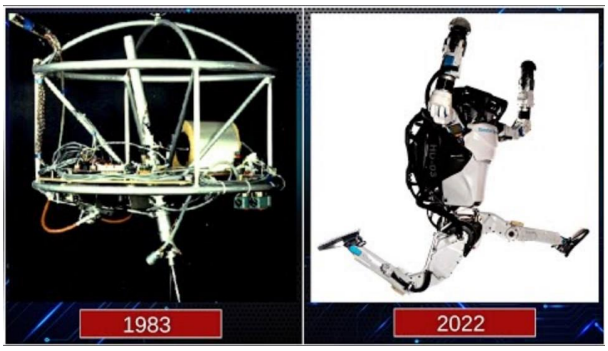
Acesse menti.com | e use o código 2295 7815

Mentimeter

"Com os avanços recentes e o crescimento explosivo das ferramentas de IA generativa, quais são suas expectativas para o impacto da IA?

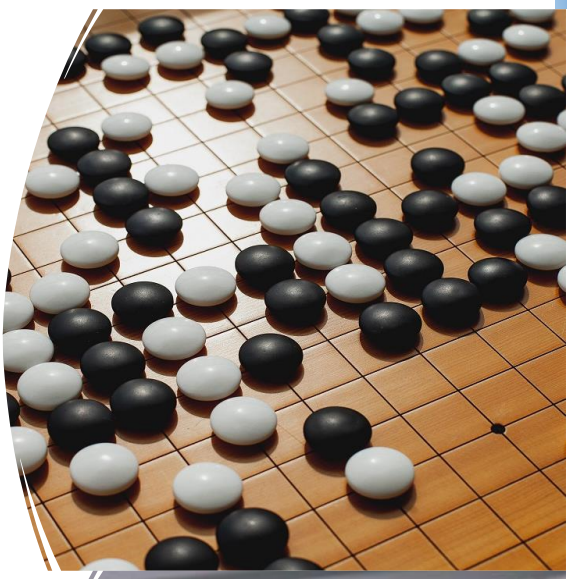
144 responses

O poder da inteligência artificial

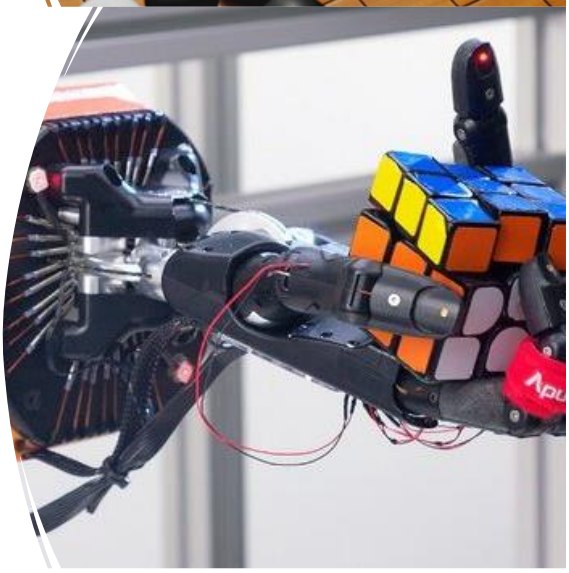


- Em 2015, um sistema de IA chamado "Eugene Goostman" conseguiu convencer 33% dos juízes humanos de que se tratava de um menino ucraniano de 13 anos, tornando-se a primeira máquina a passar no Teste de Turing. Referência: Eugene Goostman - Teste de Turing, 2015

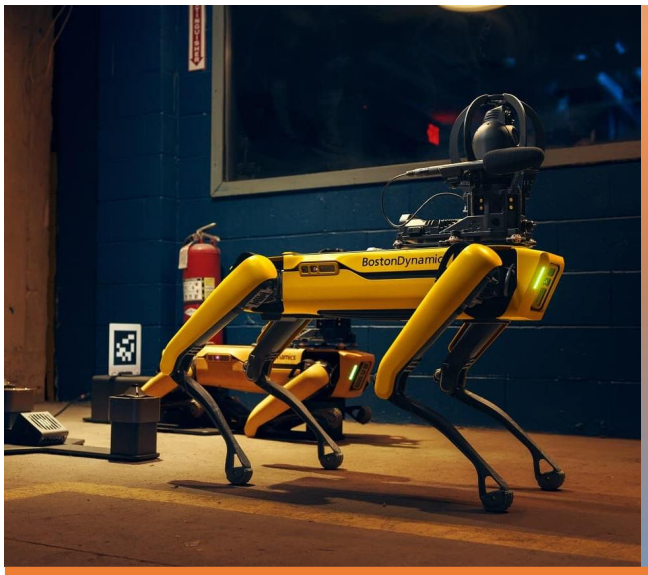
- Em 2017, pesquisadores do Google desenvolveram um sistema de IA chamado "DeepMind" que foi capaz de aprender a jogar Go em um nível sobre-humano, analisando milhares de jogos anteriores jogados por especialistas humanos. Referência: DeepMind - AlphaGo, 2017



- Também em 2017, uma equipe de pesquisadores da OpenAI desenvolveu um sistema de IA chamado "Dactyl", capaz de aprender sozinho a manipular objetos em um ambiente virtual usando uma mão robótica.
- Referência: OpenAI - Dactyl, 2017



- Em 2018, o 'Pluribus', um sistema de IA desenvolvido pelo Facebook e pela Carnegie Mellon University, derrotou jogadores profissionais em um jogo multiplayer de Texas Hold'em sem limite.
- A tomada de decisões estratégicas e a adaptabilidade da Pluribus demonstraram o potencial da IA em domínios complexos e de informação imperfeita.
- Referência: Pluribus - AI Poker, 2018



ChatGPT permite cão-robô responder perguntas

IA Benjamin

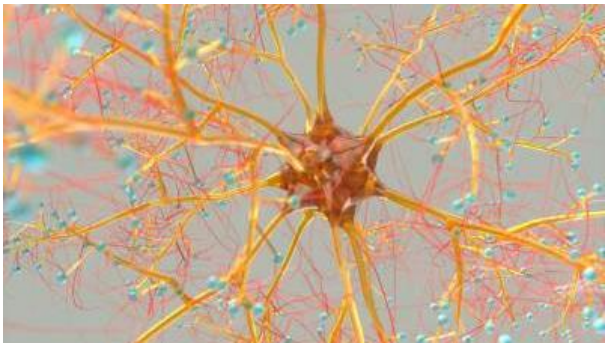
- Em 2020, um sistema de IA chamado "Benjamin" foi desenvolvido por pesquisadores do Instituto de Tecnologia de Massachusetts para escrever e produzir um curta-metragem chamado "Sunspring". O filme, que foi escrito inteiramente pelo sistema de IA, foi descrito por um crítico como "inacessível".

- https://suno.com/create

Como funciona a inteligência artificial?

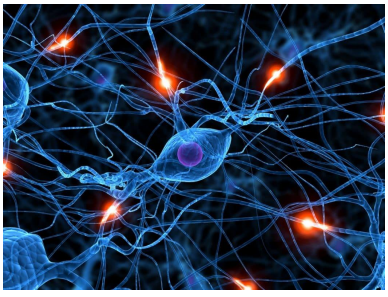
- É mágica? É bruxaria?
- Qual é a ciência por trás da aprendizagem de máquinas?

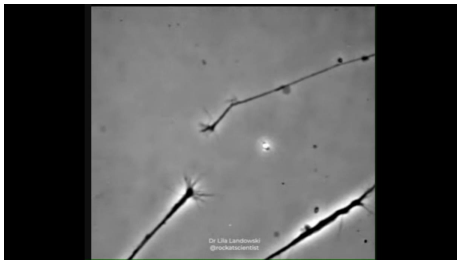
As redes neurais artificiais



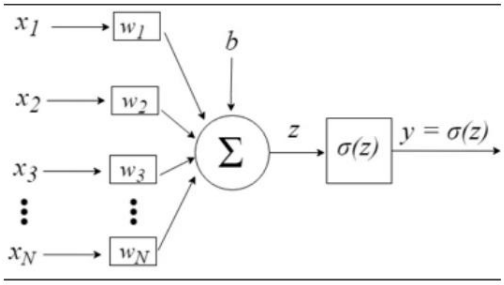
Neurônio biológico

Sinapse



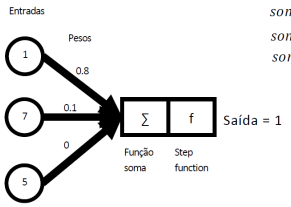


Neurônio artificial



Neurônio artificial

Neurônio artificial



$$soma = \sum_{i=1}^{n} xi * wi$$

$$soma = (1 * 0.8) + (7 * 0.1) + (5 * 0)$$

$$soma = 0.8 + 0.7 + 0$$

$$soma = 1.5$$

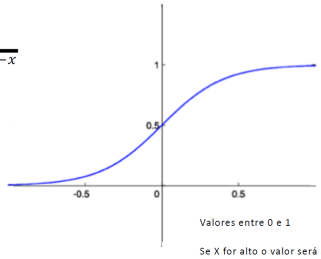
Step function (função Degrau)

Maior do que zero = 1
Caso contrário = 0

Função de ativação

Sigmoid (função sigmoide)

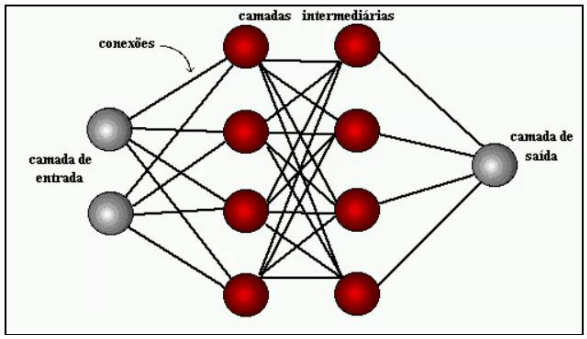
$$y = \frac{1}{1 + e^{-x}}$$

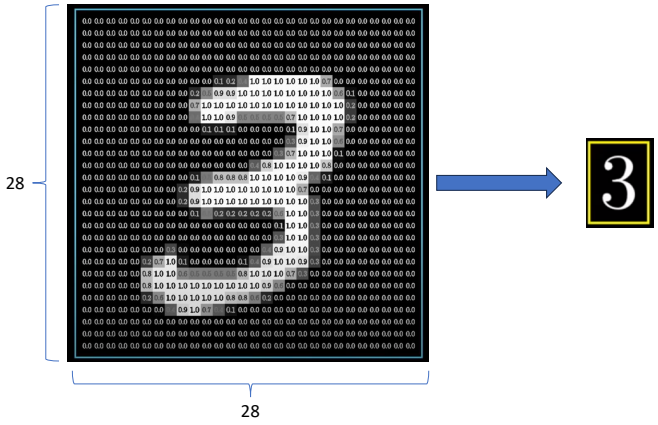


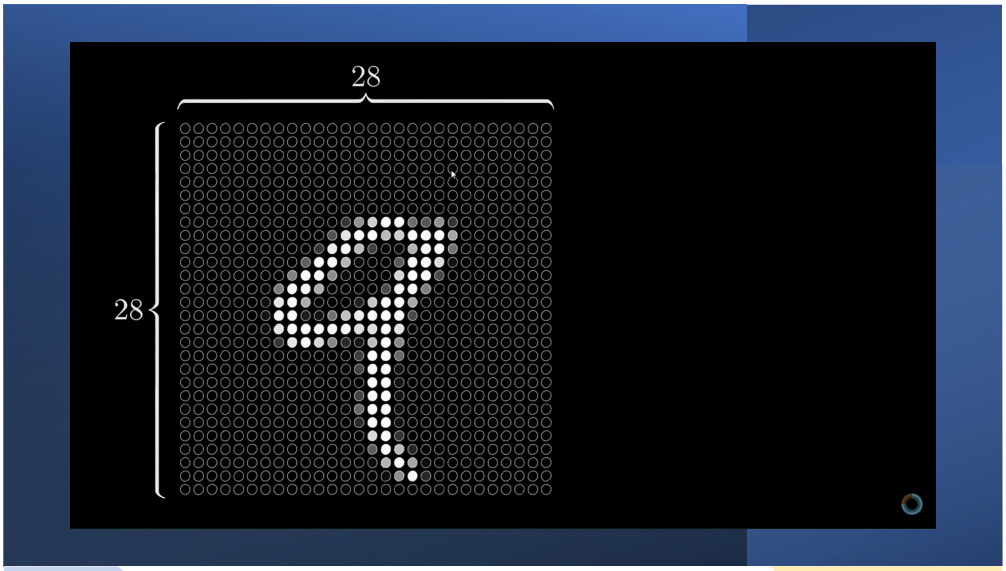
Valores entre 0 e 1

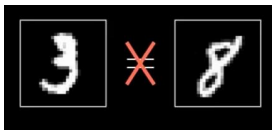
Se X for alto o valor será aproximadamente 1
Se X for pequeno o valor será aproximadamente 0
Não retorna valores negativos

Rede neural artificial e neurônio artificial

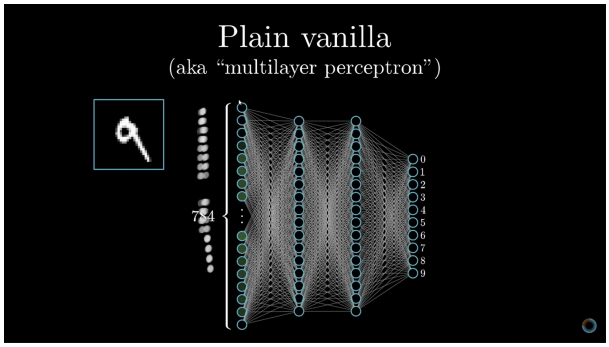




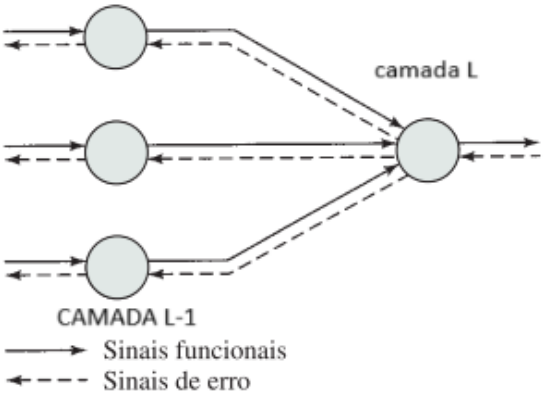




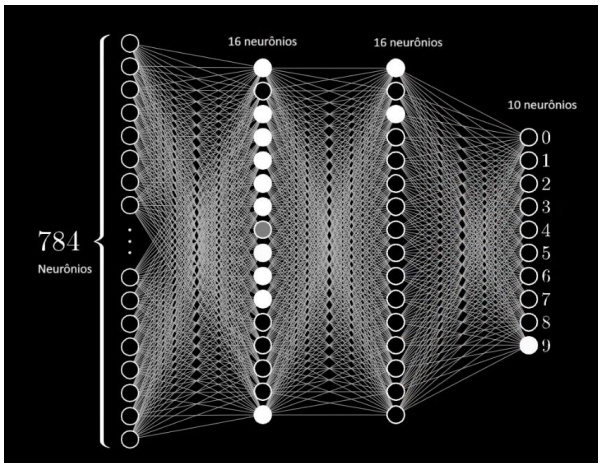
Método perceptron

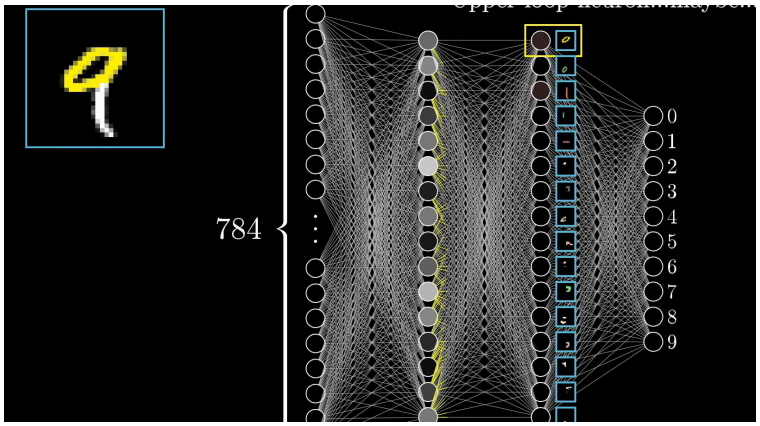


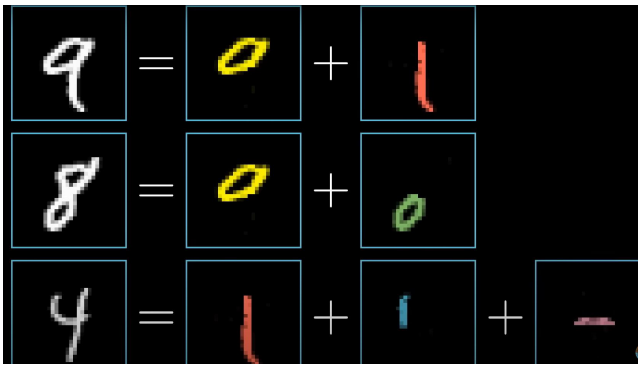
Sinais da rede



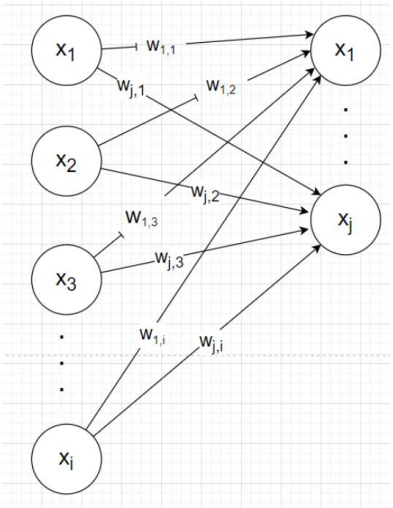
Camadas





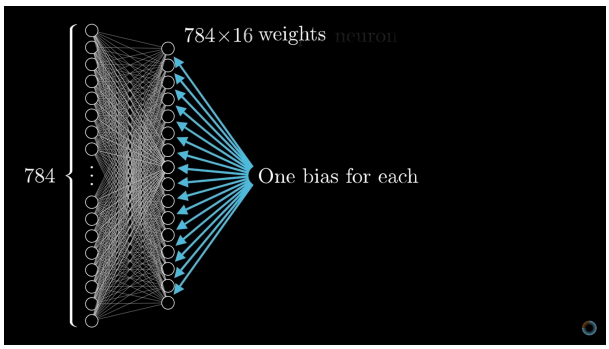


Matriz de pesos e bias

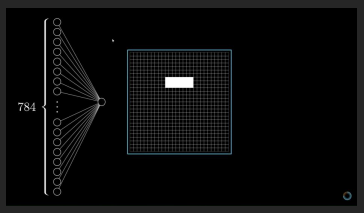


$$x_1^1 = w_{1,1}^1 \cdot x_1^0 + w_{1,2}^1 \cdot x_2^0 + w_{1,3} \cdot x_3^0 + \cdots + w_{1,784} \cdot x_{784}^0 + b_1$$

camadas



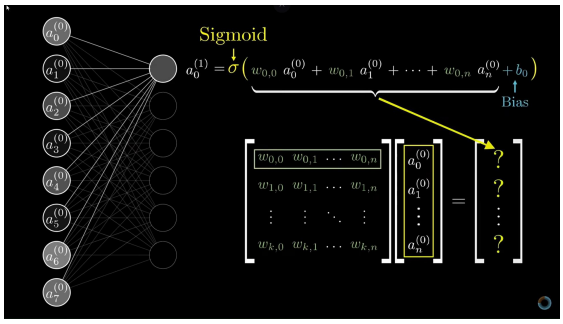
Camadas



$$x_1^1 = w_{1,1}^1 \cdot x_1^0 + w_{1,2}^1 \cdot x_2^0 + w_{1,3} \cdot x_3^0 + \cdots + w_{1,784} \cdot x_{784}^0 + b_1$$

$$\begin{bmatrix} w_{1,1}^1 & w_{1,1}^1 & \cdots & w_{1,784}^1 \\ w_{2,1}^1 & w_{2,2}^1 & \cdots & w_{2,784}^1 \\ \vdots & \ddots & \cdots & \vdots \\ w_{16,1}^1 & w_{16,2}^1 & \cdots & w_{16,784}^1 \end{bmatrix} \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_{784}^0 \end{bmatrix} + \begin{bmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_{16}^1 \end{bmatrix}$$

Rede neural



Acesse menti.com | e use o código 2295 7815

Escreva uma palavra que expressa a sua avaliação sobre o tema e a relevância da palestra.

68 responses


Avaliação