

# Aula 01 – Ementa

---

## Motivação

- Cérebro humano interpreta uma informação recebida pelos sensores como os olhos e ouvidos e associa a uma probabilidade: uma imagem pode ter 40% de chances de ser um cachorro, 15% de chances de ser um gato, 45% de ser um saco de lixo.
- Uma rede neural realiza a mesma coisa: é um dispositivo que interpreta uma informação por meio de seus neurônios.
- O processamento da rede neural se concentra em moldar uma função de probabilidade que se ajuste às entradas e saídas desejadas.
- A essência do cálculo é o cálculo da área sob uma curva: podemos encaixar quantos retângulos quisermos sob uma curva, e à medida que aumentamos o número de retângulos e a largura deles se torna cada vez menor, nos aproximamos cada vez mais da área da figura.
- Muitos problemas da engenharia e da inteligência artificial também lidam com números extremamente pequenos. Na área da otimização, muitas vezes queremos obter o valor mínimo ou máximo de uma função matemática que contenha muitos vales e montanhas. O gradiente descendente é uma das principais técnicas matemáticas para resolver estes problemas consiste em navegar pequenos passos na função, utilizando informações que indiquem a direção de maior inclinação da mesma. Estes algoritmos e técnicas precisam levar em consideração números com várias casas decimais a fim de obter uma solução de qualidade.
- A rede neural é uma técnica que necessita de otimização: obter a melhor combinação de pesos numéricos para uma estrutura de dados de forma que a diferença do erro entre um valor esperado e outro valor obtido seja o menor possível. Entretanto, usar a ideia do gradiente descendente é indesejável para este problema, devido a sua alta dimensionalidade.
- Outro problema que depende fortemente de operações numéricas são as word embeddings, ou incorporações de palavras, uma técnica da área de Processamento de Linguagem Natural, uma disciplina da Inteligência Artificial. É utilizado para transformar palavras em uma representação numérica que contenha significância semântica. Por exemplo, as palavras "homem" e "mulher", uma vez transformados em embeddings, são representados por números entre 0 e 1. Dessa forma, as operações matemáticas necessárias para transformar o número que representa "homem" no número que representa "mulher" carrega o processamento para flexão de gênero da língua portuguesa. Ao processar as frases "o homem bebeu água" e "a mulher bebeu água", as palavras

homem e mulher possuem representações numéricas muito próximas, separadas apenas pela operação de flexão de gênero. Da mesma forma, o número que representa "rei" e "rainha" também possui uma operação que é responsável pela flexão de gênero em uma frase. Além disso, os vetores que representam as operações de transformação possuem as mesmas direções, o que só reforça o papel da transformação destes vetores.

---

## A origem de tudo: álgebra linear

- Para entendermos essa transformação de palavras em word embeddings, devemos entender o que é um vetor: dado um plano cartesiano que se entende em um eixo x horizontal e um eixo y vertical, um vetor é uma seta que sempre inicia em x igual a 0 e y igual a 0 e termina em qualquer outro ponto, como por exemplo, em x igual a 1 e y igual a 2.
- Existem diversas operações matemáticas com vetores, como a soma entre vetores e a multiplicação por um escalar. Por exemplo, um vetor  $v$  que termina em x igual a 3 e y igual a -5 pode ser somado com outro vetor  $w$  que termina em x igual a 2 e y igual a 1. Somamos os valores de x que são 3 e 2, que resulta em 5, e os valores de y, que são -5 e 1, que resulta em -4, ou seja, o resultado é um vetor x igual a 5 e y igual a -4. Para a multiplicação por um escalar, podemos multiplicar um número como 2 pelo vetor  $v$  que contém x igual a 3 e y igual a -5 e obter um vetor alongado x igual a 6 e y igual a -10.
- Uma transformação linear é uma operação que utiliza uma estrutura de dados chamada matriz para transformar um vetor em outro vetor. Por exemplo, temos o vetor  $u$  que tem x igual a 5 e y igual a 7. Uma matriz de transformação 2 por 2 contém 4 números que são estes 1, -3, 2 e 4. A transformação linear consiste em multiplicarmos os dois primeiros números da matriz pelos números do vetor  $u$ , somar os produtos e guardar o resultado, assim como multiplicar os dois últimos números pelos números do vetor  $u$ , somar os produtos e guardar o resultado. Os dois resultados compoem o vetor transformado. Assim, multiplicamos 1 vezes 5 e -3 vezes 7, que resultam em 5 e -21, que somados resultam em -16, que é o valor de x do vetor transformado. De maneira análoga, multiplicamos 2 vezes 5 e 4 vezes 7, que resultam em 10 e 28, que somados resultam em 38, que é o valor de y do vetor transformado. Assim, usamos a matriz especificada para transformar o vetor  $u$  com x igual a 5 e y igual a 7 em um vetor x igual a -16 e y igual a 38. Essa é a mesma ideia que usamos nas word embeddings: a palavra original sofre uma transformação linear através de uma matriz e se torna um outro vetor. A matriz deve conter os números que realizam uma flexão de gênero, de número, de verbo, etc.
- Outra operação importante entre vetores é o produto escalar (ou "dot product" em inglês). Seja um vetor  $v$  com x igual a 4 e y igual a 1 e um vetor  $w$  com x igual a 2 e y igual a -1, o produto vetorial multiplica os valores de x entre os vetores, assim como os valores de y, e soma os resultados. Assim, o exemplo resulta em 4 vezes 2 igual a 8 e 1 vezes -1 igual a -1, onde 8 somado com -1 resulta em 7. Como o resultado é positivo,  $v$  e  $w$  apontam em direções semelhantes. Caso

resultasse em zero, os vetores seriam perpendiculares, com um ângulo de 90 graus entre eles. Se o valor fosse negativo, os vetores apontariam em direções opostas. Dessa forma, o produto escalar mede o quanto dois vetores estão alinhados.

---

# IA e matemática: alguns exemplos

## Redes neurais

- Os modelos de inteligência artificial, dos mais simples aos mais modernos, podem ser compreendidos como transformações matemáticas que convertem uma entrada em uma saída significativa. Em essência, uma rede neural aprende a mapear dados de entrada para respostas desejadas. Por exemplo, um modelo de classificação de imagens de cães transforma uma foto contendo padrões visuais, como o formato das orelhas, a coloração da pelagem e o tamanho do focinho, em uma saída que representa a raça do animal, como Blenheim Spaniel. De modo semelhante, um modelo de geração de texto, como o GPT, Gemini ou Deepseek, transforma a sequência de palavras "A Senhora Maria estava parada, mas agora ela avançou..." em outra sequência coerente, prevendo palavra por palavra, "um", "passo", "em", "direção", e assim por diante, até formar uma frase completa e gramaticalmente correta.
- Em termos matemáticos, cada neurônio de uma rede neural realiza uma transformação linear seguida de uma não linearidade. A equação fundamental que gera a saída de um neurônio consiste no produto escalar entre o vetor de entrada e o vetor de pesos, este último representando o conhecimento adquirido durante o treinamento. O resultado é somado a um bias, um termo adicional que desloca o resultado da operação e confere maior flexibilidade à rede. Essa soma é então passada para uma função de ativação, que introduz comportamento não linear e permite que o modelo aprenda relações complexas entre variáveis, indo além de simples combinações lineares.
- A determinação dos pesos e dos bias ocorre na etapa de treinamento, onde o algoritmo de gradiente descendente ajusta os parâmetros para minimizar o erro entre a saída prevista pelo modelo e a saída real desejada. Esse processo envolve o cálculo de derivadas parciais, gerando assim o gradiente que indica a direção e a intensidade da mudança necessária em cada parâmetro para reduzir o erro. Trata-se, portanto, de uma aplicação direta de cálculo diferencial e otimização numérica.
- Um exemplo simples dessa lógica pode ser observado na regressão linear, um dos modelos mais básicos de aprendizado de máquina. Dada uma coleção de pontos em um plano cartesiano, o objetivo é ajustar uma reta que melhor se aproxime dos dados, minimizando a soma dos erros verticais entre os pontos observados e os valores previstos pela reta. Para isso, dois parâmetros são ajustados: a inclinação e o intercepto da reta. Assim como nas redes neurais, o gradiente descendente é utilizado para encontrar a combinação desses parâmetros que produz o menor

erro possível, mostrando como os mesmos fundamentos matemáticos sustentam tanto os modelos clássicos quanto as arquiteturas modernas de inteligência artificial.

## Aprendizado por reforço

- O aprendizado por reforço consiste em um processo de decisão de Markov no qual um agente interage com um ambiente, executando ações que resultam em recompensas positivas ou negativas. Cada interação gera um novo estado do ambiente, e o agente utiliza essas informações para aprender políticas, ou estratégias de ação, que maximizem o ganho acumulado de recompensas ao longo do tempo. O objetivo final é encontrar o melhor conjunto de políticas capazes de guiar o agente a tomar decisões que, em média, levem aos melhores resultados possíveis.
- Por exemplo, imagine um jogo de labirinto em que o agente pode se mover em quatro direções (cima, baixo, esquerda e direita). A cada movimento, ele recebe uma recompensa positiva ao se aproximar da saída e uma negativa ao colidir com uma parede ou seguir um caminho errado. O desafio é descobrir, por tentativa e erro, qual sequência de movimentos leva o jogador à saída de forma mais eficiente, considerando as barreiras e penalidades ao longo do percurso.
- Esse tipo de abordagem é amplamente utilizado em problemas de tomada de decisão sequencial, como controle de robôs, jogos estratégicos, sistemas de recomendação e veículos autônomos. Por trás do aprendizado por reforço estão conceitos fundamentais da matemática computacional, como otimização, teoria das probabilidades e cálculo de expectativas, que permitem formalizar o processo de aprendizado e definir as melhores estratégias possíveis para cada situação.
- Outros exemplos:
- Algoritmo para jogar xadrez e shogi que usa aprendizado por reforço ao jogar consigo mesmo
- Planejamento de modelos para jogar Atari, Go, Xadrez e Shogi
- Otimização de prefixos paralelos de circuitos usando aprendizado por reforço
- Controle magnético de plasmas tokamak com aprendizado por reforço
- Ajuste automática de operações de turbinas de gasolina

## Otimização

- O algoritmo de Otimização por Enxame de Partículas (do inglês, Particle Swarm Optimization - PSO) utiliza uma notação vetorial para representar um conjunto de partículas, cada uma correspondendo a uma solução candidata dentro do espaço de busca. Inspirado no comportamento coletivo de bandos de pássaros e cardumes de peixes, o PSO faz com que essas

partículas naveguem por uma função, ajustando continuamente suas posições e velocidades para encontrar o ponto de mínimo ou máximo desejado.

- Durante o processo, cada partícula compartilha e atualiza informações baseadas em três componentes principais:
  1. A melhor posição individual, que representa o ponto onde aquela partícula obteve o melhor resultado até o momento.
  2. A melhor posição global, correspondente à melhor posição já encontrada por todo o enxame.
  3. O termo de inércia, que expressa a tendência da partícula de manter parte de seu movimento anterior, equilibrando a exploração de novas regiões do espaço de busca e a exploração local de soluções promissoras.
- Esses três vetores são combinados em uma equação que atualiza iterativamente a direção e a intensidade do movimento de cada partícula. Assim, o PSO exemplifica de forma elegante como conceitos matemáticos vetoriais, álgebra linear e otimização numérica podem se unir para resolver problemas complexos, explorando o espaço de soluções de maneira cooperativa e adaptativa.

## Transformers

- Transformers são a base para os modelos de linguagem de larga escala como o GPT, Gemini e Deepseek. A ideia geral consiste em processar sequências de dados, como palavras em um texto, utilizando mecanismos de atenção que ponderam a importância relativa de cada elemento da sequência em relação aos demais. Em vez de tratar as palavras de forma isolada ou em janelas fixas, os transformers empregam atenção auto-regressiva para capturar relações de longo alcance entre os termos, permitindo que o modelo aprenda dependências complexas e contextos sutis dentro de grandes volumes de dados textuais.
- A combinação entre word embeddings e os modelos transformers torna possível a compreensão de elementos semânticos dentro das frases alimentadas, dependendo do contexto. Por exemplo, a palavra "modelo" possui significados diferentes nas frases "um modelo de machine learning" e "um modelo de moda". A estrutura do transformer permite capturar essas diferenças. Essa capacidade decorre de operações matemáticas fundamentais, como produtos escalares entre vetores de embeddings e o uso de softmax para normalizar pesos de atenção, técnicas de álgebra linear e cálculo numérico que estão no cerne da matemática computacional. Assim, os transformers representam um exemplo claro de como conceitos matemáticos aplicados sustentam os avanços mais sofisticados da inteligência artificial moderna.

## Ciência de dados

- Existem diversos processos de gerenciamento de projetos de ciência e mineração de dados, que definem metodologias estruturadas para conduzir uma análise de forma sistemática, reprodutível e orientada a resultados. O mais conhecido desses processos é o KDD (do inglês, Knowledge Discovery in Databases, ou Descoberta de Conhecimento em Bases de Dados), um modelo conceitual que descreve as etapas necessárias para extrair conhecimento útil e compreensível a partir de grandes volumes de dados.
  - Possui 5 etapas que consistem na seleção dos dados a serem explorados, o pré-processamento dos dados de forma a aumentar a qualidade da análise, a transformação dos dados para um formato adequado para as ferramentas de análise, a mineração de dados onde são aplicadas as técnicas de descoberta de conhecimento (que podem incluir ou não aprendizado de máquina) e a interpretação dos resultados onde serão expostas as métricas que definem a qualidade da análise e que permitem o ajuste das etapas anteriores.
  - O KDD serve de base para metodologias mais recentes, como o CRISP-DM (do inglês, Cross Industry Standard Process for Data Mining, ou Processo Padrão Cruzado para Mineração de Dados), que ampliou e padronizou o processo para aplicações em diferentes domínios. Ambos os modelos destacam o papel essencial da matemática computacional, especialmente nas fases de transformação, mineração e avaliação, como o alicerce teórico e prático da descoberta de conhecimento em bases de dados.
- 

## Ementa da disciplina:

Vamos ter 4 principais assuntos na disciplina

1. Noções básicas sobre erros: estuda os diferentes tipos de erros numéricos que ocorrem em cálculos computacionais, como erros de arredondamento e truncamento, e suas implicações na precisão e estabilidade dos resultados.
  2. Zeros de funções reais: aborda métodos numéricos para encontrar as raízes de equações não lineares, ou seja, os valores de  $x$  que satisfazem uma função  $f$  de  $x$  igual a 0, utilizando métodos iterativos como o de Newton-Raphson.
  3. Resolução de sistemas lineares: explora métodos diretos e iterativos para resolver sistemas de equações lineares do tipo  $A \text{ vezes } x \text{ igual a } b$ , fundamentais em diversas áreas da computação científica, como o método de Gauss e o método de Jacobi.
  4. Otimização: estuda técnicas para determinar o ponto que minimiza ou maximiza uma função, com ou sem restrições, aplicando métodos analíticos e numéricos utilizados em problemas de engenharia, ciência de dados e inteligência artificial.
-

# Ferramentas utilizadas na disciplina

- Jupyter Notebook: ambiente interativo que combina código, texto e visualizações em um único documento, ideal para experimentação, ensino e documentação de projetos em ciência de dados e matemática computacional.
- Python: linguagem de programação de alto nível, versátil e de fácil leitura, amplamente utilizada em computação científica, análise de dados e inteligência artificial devido à sua vasta biblioteca de ferramentas numéricas e estatísticas.
- Git: sistema de controle de versão distribuído que permite registrar o histórico de alterações em projetos, facilitar o trabalho colaborativo e garantir rastreabilidade e segurança no desenvolvimento de código e documentação.