



## UNIDADE III

---

### Inteligência Artificial

Profa. Dra. Vanessa Lessa

# Aprendizagem de máquina

- Aprendizagem de máquina (*Machine Learning*) é uma subárea da inteligência artificial que se concentra em desenvolver técnicas e algoritmos que permitem que as máquinas aprendam com dados, sem serem explicitamente programadas. Aprendizagem de máquina é dividida em três categorias principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.
- O aprendizado de máquina manipula uma grande quantidade de dados para realizar o treinamento de um sistema computacional, possibilitando que o sistema reconheça determinados padrões dentro de um grupo de dados. Os métodos utilizados geralmente são complexos, dessa forma é muito importante o planejamento, a análise e a preparação dos dados (BISHOP, 2006).

# Aprendizagem de máquina

- Aprendizado supervisionado é quando um algoritmo é treinado com dados rotulados, ou seja, dados que já têm uma classificação ou saída desejada. O objetivo é fazer com que o algoritmo aprenda a generalizar essas saídas para novos dados não vistos anteriormente. Exemplos de técnicas de aprendizado supervisionado incluem regressão linear, árvores de decisão e redes neurais.
- Aprendizado não supervisionado é quando um algoritmo é treinado com dados não rotulados. O objetivo é explorar os dados e encontrar padrões ou estruturas escondidas neles. Exemplos de técnicas de aprendizado não supervisionado incluem *clustering*, detecção de *outliers* e redução de dimensionalidade.
  - Aprendizado por reforço é quando um algoritmo é treinado por meio de uma série de ações e recompensas. O objetivo é fazer com que o algoritmo aprenda a tomar ações que maximizem a recompensa. Exemplos de técnicas de aprendizado por reforço incluem Q-learning e algoritmos de política ótima.

# Aprendizado supervisionado

- Existem dois tipos de tarefas de aprendizado supervisionado: classificação e regressão. Classificação é quando o algoritmo é treinado para atribuir dados a uma das várias categorias possíveis. Por exemplo, classificar as imagens de animais em cães ou gatos. Regressão é quando o algoritmo é treinado para prever um valor contínuo, como o preço de uma casa ou a temperatura futura.
- O processo de aprendizado supervisionado é composto de três etapas: treinamento, validação e teste.
  - Na etapa de treinamento, o algoritmo é exposto aos dados rotulados e ajusta seus parâmetros para minimizar o erro entre as saídas previstas e as saídas reais.
    - Na etapa de validação, o algoritmo é avaliado em um conjunto de dados separado para garantir que está generalizando corretamente e não está se ajustando excessivamente aos dados de treinamento.
    - Na etapa de teste, o algoritmo é avaliado em um conjunto de dados ainda não visto para avaliar sua capacidade de generalizar para novos dados.

## Exemplo: abordagem com base em árvores de decisão

- Abordagem com base em árvores de decisão é uma técnica de aprendizado supervisionado de máquina que se baseia em dividir um conjunto de dados em subconjuntos cada vez menores por meio da aplicação de regras de decisão.
- Essas regras são aplicadas a cada nó da árvore, desde a raiz até a folha, que contém a classificação ou a decisão final.
- O objetivo é encontrar regras que melhor se ajustem aos dados e permitam uma classificação precisa.
- A construção de uma árvore de decisão é feita em duas etapas: geração e poda.
  - Na etapa de geração, a árvore é construída a partir do conjunto de treinamento. Isso é feito dividindo o conjunto de dados em subconjuntos cada vez menores por meio da aplicação de regras de decisão. Cada subconjunto é então associado a uma folha que contém a classificação ou a decisão final. Na etapa de poda, a árvore é simplificada para evitar *overfitting*. Isso é feito removendo nós que não contribuem para a precisão da classificação.

## Exemplo: abordagem com base em árvores de decisão

- Uma das principais vantagens da abordagem com base em árvores de decisão é a facilidade de interpretação. A árvore resultante é fácil de entender e interpretar, o que é útil para tomar decisões. Além disso, as árvores de decisão são robustas a *outliers* e não requerem muitos pressupostos sobre os dados. No entanto, elas podem ser propensas a *overfitting* se não forem cuidadosamente podadas.

# Algoritmo ID3

- O algoritmo ID3 (Iterative Dichotomiser 3) é um algoritmo de aprendizado supervisionado de árvores de decisão desenvolvido por Ross Quinlan em 1986. Ele é usado para construir uma árvore de decisão a partir de um conjunto de dados rotulados. O objetivo é encontrar as regras de decisão que melhor se ajustem aos dados e permitam uma classificação precisa.

# Algoritmo ID3

O algoritmo ID3 funciona da seguinte maneira:

1. Escolha a melhor característica para dividir o conjunto de dados atual. Isso é feito calculando a medida de ganho de informação para cada característica e escolhendo a característica com o maior ganho.
2. Divida o conjunto de dados atual de acordo com os valores da característica escolhida. Crie um nó para cada valor da característica e associe o subconjunto de dados correspondente a esse nó.
3. Repita os passos 1 e 2 para cada subconjunto de dados até que todos os nós sejam folhas ou até que a precisão desejada seja alcançada.
4. O algoritmo ID3 é eficiente e escalável para grandes conjuntos de dados. No entanto, ele é propenso a *overfitting* se o conjunto de dados contiver ruído ou se as regras de decisão forem muito complexas.



# Aprendizado não supervisionado

- Aprendizado não supervisionado é uma categoria de aprendizado de máquina em que um algoritmo é treinado com dados não rotulados. O objetivo é explorar os dados e encontrar padrões ou estruturas escondidas neles. Exemplos de técnicas de aprendizado não supervisionado incluem *clustering*, detecção de *outliers* e redução de dimensionalidade.
- *Clustering* é uma técnica de agrupamento em que os algoritmos tentam encontrar padrões ou estruturas nos dados, agrupando-os em conjuntos de dados similares.
- A detecção de *outliers* é uma técnica de detecção de anomalia que busca identificar pontos de dados que são significativamente diferentes dos demais.
  - A redução de dimensionalidade é uma técnica usada para reduzir a complexidade dos dados, removendo redundâncias e mantendo a maior parte da informação importante.

## Exemplo: clusterização conceitual

- A clusterização conceitual é uma técnica de aprendizado não supervisionado que se concentra em encontrar padrões ou estruturas escondidas em dados não rotulados por meio do agrupamento de dados baseado em sua similaridade semântica. Ela é diferente das técnicas de clusterização tradicionais, como k-means e agrupamento hierárquico, que se baseiam apenas na similaridade dos dados.
- A clusterização conceitual se baseia na ideia de que os dados podem ser agrupados de acordo com seu significado semântico, e não apenas com base na similaridade dos dados. Por exemplo, dois documentos podem ser semelhantes em termos de sua estrutura, mas diferentes em termos de seu conteúdo. A clusterização conceitual tenta agrupar esses documentos de acordo com seu conteúdo, e não apenas com base na estrutura dos dados.

## Exemplo: clusterização conceitual

- A clusterização conceitual é realizada em três etapas principais: pré-processamento, agrupamento e pós-processamento.
- Pré-processamento: os dados são limpos e preparados para o agrupamento. Isso inclui a remoção de ruído, a normalização dos dados e a remoção de palavras irrelevantes. Além disso, é importante realizar a representação dos dados, como a utilização de técnicas de processamento de linguagem natural para criar vetores de recursos.
- Agrupamento: os dados são agrupados de acordo com sua similaridade semântica. Isso é feito utilizando algoritmos de clusterização conceitual. Esses algoritmos buscam encontrar tópicos ocultos nos dados e agrupar os dados de acordo com esses tópicos.
  - Pós-processamento: os resultados são analisados e interpretados. Isso inclui a avaliação da qualidade dos *clusters* gerados, a interpretação dos tópicos encontrados e a identificação de relações entre os *clusters*. Além disso, é possível realizar a etapa de rotulação dos *clusters*, atribuindo-lhes nomes que representem o conteúdo dos elementos agrupados.

# COBWEB

- COBWEB é um algoritmo de clusterização conceitual desenvolvido por Andrew Moore em 1993. Ele é baseado no algoritmo de classificação hierárquica e é projetado para agrupar dados de acordo com sua similaridade semântica.
- O COBWEB é capaz de lidar com dados categóricos e numéricos, e é projetado para lidar com dados com uma estrutura hierárquica. Além disso, é capaz de lidar com dados incompletos e ruidosos, e é escalável para grandes conjuntos de dados. O COBWEB é usado em uma variedade de aplicações, como recuperação de informações, processamento de linguagem natural e análise de mercado.
  - O COBWEB é uma técnica de agrupamento hierárquico, ou seja, ele gera uma estrutura de hierarquia de *clusters* e não apenas um conjunto de *clusters* finais. Além disso, é importante definir critérios para a parada do algoritmo, pois ele pode continuar agrupando *clusters* até que não haja mais dados para serem agrupados.

# COBWEB

O algoritmo COBWEB funciona da seguinte maneira:

1. Inicialmente, cada instância é considerada como um *cluster* individual.
2. Encontra-se a instância mais semelhante ao *cluster* atual.
3. As instâncias são agrupadas em um novo *cluster*.
4. A similaridade entre os novos *clusters* é calculada por meio de um critério de similaridade, como distância de Jaccard.
5. Se a similaridade entre os *clusters* é maior que um limiar predefinido, os *clusters* são fundidos.
6. Esse processo é repetido até que não haja nenhuma fusão possível ou até que o número de *clusters* alcance o valor desejado.

# Interatividade

Analise as afirmações:

- I. Aprendizagem de máquina se concentra em desenvolver técnicas e algoritmos para que as máquinas aprendam com dados, sem serem explicitamente programadas.
- II. Aprendizado supervisionado é quando um algoritmo é treinado com dados não rotulados, ou seja, dados que já têm uma classificação ou saída desejada.
- III. Aprendizado não supervisionado é quando um algoritmo é treinado com dados rotulados.
- IV. Aprendizado por reforço é quando um algoritmo é treinado por meio de uma série de ações e recompensas.
- V. Existem dois tipos de tarefas de aprendizado supervisionado: classificação e regressão.

Estão corretas:

- a) I, II e III.
- b) III e IV.
- c) I, II e V.
- d) I, IV e V.
- e) Todas as afirmações.

# Resposta

Analise as afirmações:

- I. Aprendizagem de máquina se concentra em desenvolver técnicas e algoritmos para que as máquinas aprendam com dados, sem serem explicitamente programadas.
- II. Aprendizado supervisionado é quando um algoritmo é treinado com dados não rotulados, ou seja, dados que já têm uma classificação ou saída desejada.
- III. Aprendizado não supervisionado é quando um algoritmo é treinado com dados rotulados.
- IV. Aprendizado por reforço é quando um algoritmo é treinado por meio de uma série de ações e recompensas.
- V. Existem dois tipos de tarefas de aprendizado supervisionado: classificação e regressão.

Estão corretas:

- a) I, II e III.
- b) III e IV.
- c) I, II e V.
- d) I, IV e V.
- e) Todas as afirmações.

# Aprendizagem por reforço

- Aprendizagem por reforço é uma categoria de aprendizado de máquina em que um agente aprende a tomar ações apropriadas em um ambiente por meio da experimentação e do recebimento de recompensas ou punições. O objetivo é maximizar a recompensa acumulada ao longo do tempo.
- A aprendizagem por reforço é baseada na ideia de que um agente pode aprender a tomar ações apropriadas por meio da tentativa e erro. O agente é colocado em um ambiente e é recompensado ou punido de acordo com suas ações. Ele então usa essas recompensas ou punições para ajustar sua estratégia e melhorar suas ações futuras.



# Elementos básicos da aprendizagem por reforço

Existem alguns elementos básicos que compõem a aprendizagem por reforço:

- Agente: é a entidade que toma ações no ambiente. Ele é responsável por selecionar e executar ações com base em suas crenças sobre o estado do ambiente.
- Ambiente: é o contexto no qual o agente age. Ele fornece informações sobre o estado atual do sistema e responde às ações do agente com recompensas ou punições.
- Ação: é a escolha feita pelo agente para interagir com o ambiente. Ele pode ser uma ação discreta ou contínua.
  - Recompensa: é o *feedback* positivo fornecido pelo ambiente em resposta às ações do agente. É utilizado pelo agente para avaliar o sucesso de suas ações e tomar decisões futuras.

# Elementos básicos da aprendizagem por reforço

- Política: é a estratégia utilizada pelo agente para selecionar ações. Ele pode ser uma função que mapeia estados para ações ou uma distribuição de probabilidade sobre ações.
- Função de valor: é a avaliação do agente sobre o valor de cada estado ou ação. Ele é utilizado pelo agente para avaliar o sucesso de suas ações e tomar decisões futuras.
- Algoritmo de aprendizagem: é o mecanismo utilizado pelo agente para aprender a partir das recompensas e punições. Ele pode ser baseado em valor, política ou híbrido.

## Exemplo: jogo da velha

- Jogo da velha é um jogo simples para duas pessoas. O jogo é jogado em uma grade 3x3 e cada jogador tem uma marcação, geralmente um "X" ou um "O". O objetivo do jogo é ser o primeiro jogador a conseguir colocar três de suas marcações em uma linha horizontal, vertical ou diagonal.
- Em inteligência artificial, o jogo da velha pode ser usado como um problema de teste para algoritmos de busca, aprendizado de máquina e inteligência artificial. O jogo pode ser jogado de forma completamente aleatória, em que o computador escolhe uma posição livre aleatoriamente, ou usando algoritmos mais sofisticados, como algoritmos min-max ou redes neurais.

## Exemplo: jogo da velha

- Algoritmos min-max são algoritmos de busca em árvore que é usado para encontrar a jogada ótima para o jogador atual, levando em consideração todas as possíveis jogadas futuras e seus resultados esperados. Eles maximizam a recompensa do jogador atual, ao mesmo tempo em que minimizam a recompensa do jogador adversário.
- Redes neurais são usadas para aprender a jogar o jogo da velha, treinando-as com exemplos de jogadas vitoriosas e perdedoras, o que permite que o computador faça previsões precisas sobre a melhor jogada a ser feita em uma determinada situação.
  - O jogo da velha é considerado um problema simples, mas pode se tornar desafiador quando se adiciona uma camada de inteligência artificial a ele. Além disso, ele também pode ser usado como uma metáfora para outros problemas mais complexos, como jogos de tabuleiro e problemas de tomada de decisão em negócios etc.

# Aplicando realidade aumentada (AR) ao jogo da velha

- A inteligência artificial (IA) é um campo da computação que se concentra na criação de sistemas que imitam o comportamento inteligente humano. A IA tem muitas aplicações, incluindo jogos. Um jogo popular para testar e desenvolver algoritmos de IA é o jogo da velha.
- A realidade aumentada (AR) é uma tecnologia que combina informações virtuais e reais para criar uma experiência de jogo mais imersiva. Ao aplicar AR ao jogo da velha, é possível transformar o tabuleiro em uma tela virtual que pode ser jogada em qualquer lugar e permitir que os jogadores vejam a jogada do outro jogador em tempo real.
  - A inteligência artificial pode ser aplicada ao jogo da velha de várias maneiras, como usando algoritmos de busca e redes neurais, e a realidade aumentada pode ser usada para melhorar a experiência de jogo.

# Redes neurais

- As redes neurais são uma técnica de inteligência artificial baseada na estrutura do cérebro humano. Elas consistem em camadas de nós ou "neurônios" conectados entre si, que trabalham juntos para realizar tarefas específicas, como reconhecimento de imagens, processamento de linguagem natural e tomada de decisão.
- Existem vários tipos de redes neurais, incluindo redes neurais *feedforward*, que passam informações de entrada por meio de camadas sucessivas de neurônios sem voltar para trás, e redes neurais recorrentes, que permitem informações de estado retornarem à rede.
  - Aprendizado de máquina é a forma como as redes neurais aprendem a realizar tarefas específicas. Isso é feito por meio de um processo de treinamento, em que a rede é exposta a um grande conjunto de exemplos de entrada e saída, e os pesos dos conectores entre os neurônios são ajustados para minimizar a diferença entre a saída desejada e a saída produzida pela rede.

# O neurônio artificial

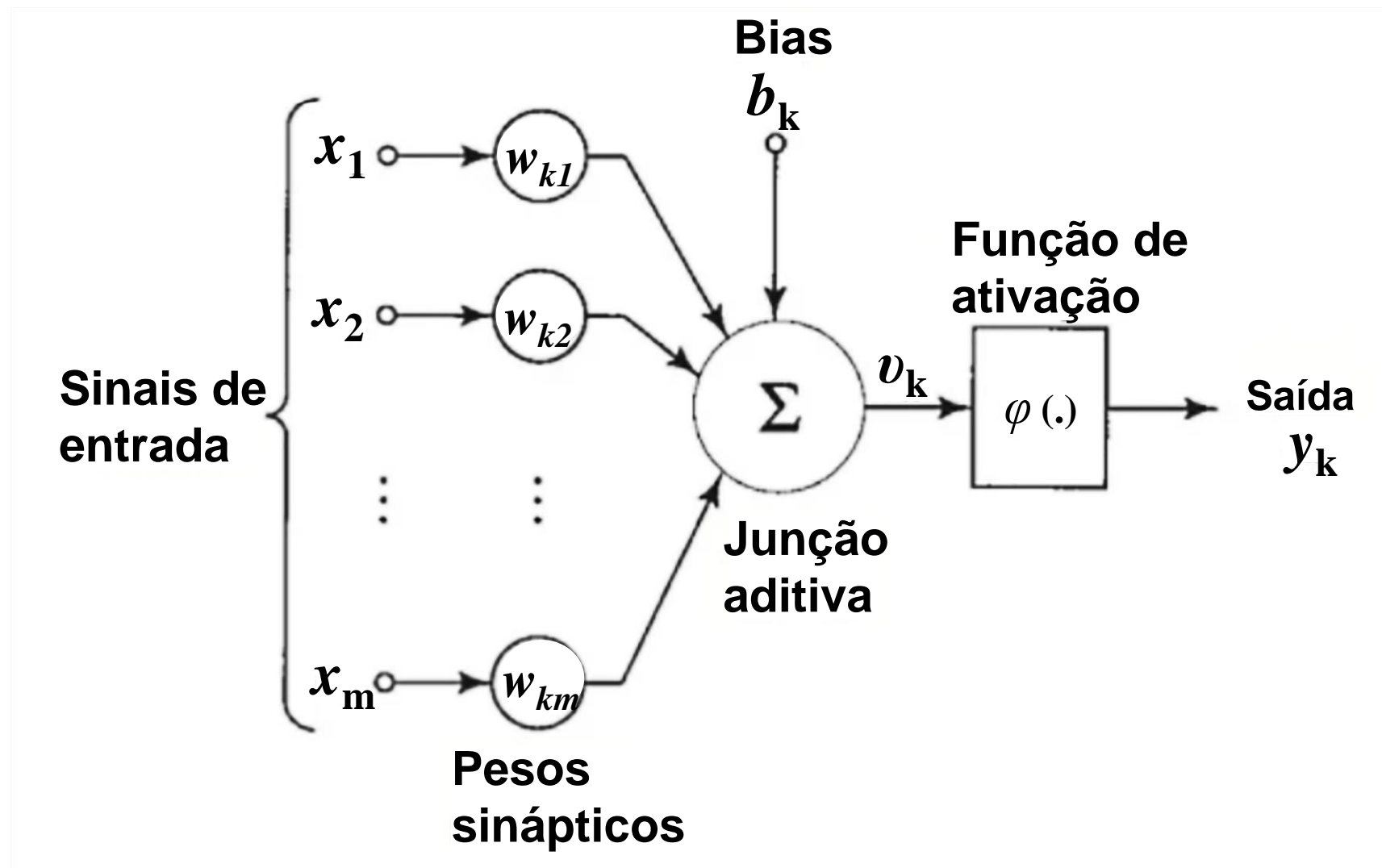
- Um neurônio artificial é uma unidade básica de uma rede neural artificial, que é projetada para simular a estrutura e função de um neurônio biológico. Cada neurônio artificial tem entradas, saídas e pesos que conectam suas entradas às suas saídas.
- A operação básica de um neurônio artificial é a de realizar uma soma ponderada das entradas e aplicar uma função de ativação a esse resultado. A função de ativação é uma função matemática que determina se o neurônio está "ativado" ou "inativo", ou seja, se ele irá transmitir uma saída ou não (HAYKIN, 1994).
  - Os pesos dos neurônios são valores numéricos que determinam a importância relativa das entradas para a saída. Durante o processo de treinamento, os pesos dos neurônios são ajustados para minimizar a diferença entre a saída desejada e a saída produzida pela rede.

# O neurônio artificial

- Os neurônios artificiais são organizados em camadas, em que cada camada é composta por vários neurônios. A primeira camada é chamada de camada de entrada e é responsável por receber os dados de entrada. A última camada é chamada de camada de saída e é responsável por produzir a saída da rede. As camadas intermediárias são chamadas de camadas ocultas e têm a função de realizar operações complexas e extrair características importantes dos dados de entrada.
- O neurônio artificial é a unidade básica da rede neural artificial, que é responsável por realizar as operações matemáticas e tomar decisões. É uma abstração da estrutura e função do neurônio biológico, que é usada para simular a capacidade de aprendizado e adaptação do cérebro humano.



# Modelo de neurônio artificial



# Interatividade

Analise as afirmações:

- I. A aprendizagem por reforço é baseada na ideia de que um agente pode aprender a tomar ações apropriadas por meio da tentativa e erro.
- II. A realidade aumentada (AR) é uma tecnologia que combina informações virtuais e reais para criar uma experiência de jogo mais imersiva.
- III. As redes neurais são uma técnica de IA baseada na estrutura do cérebro humano.
- IV. Um neurônio artificial é uma unidade básica de uma rede neural artificial.
- V. Os neurônios artificiais são organizados em camadas, em que cada camada é composta por vários neurônios.

Estão corretas:

- a) I, II e III.
- b) I, II e IV.
- c) I, II e V.
- d) I, III, IV e V.
- e) Todas as afirmações.

# Resposta

Analise as afirmações:

- I. A aprendizagem por reforço é baseada na ideia de que um agente pode aprender a tomar ações apropriadas por meio da tentativa e erro.
- II. A realidade aumentada (AR) é uma tecnologia que combina informações virtuais e reais para criar uma experiência de jogo mais imersiva.
- III. As redes neurais são uma técnica de IA baseada na estrutura do cérebro humano.
- IV. Um neurônio artificial é uma unidade básica de uma rede neural artificial.
- V. Os neurônios artificiais são organizados em camadas, em que cada camada é composta por vários neurônios.

Estão corretas:

- a) I, II e III.
- b) I, II e IV.
- c) I, II e V.
- d) I, III, IV e V.
- e) Todas as afirmações.

# Treinamento de um perceptron

- O perceptron é uma arquitetura simples de rede neural artificial, composta por uma única camada de neurônios, que é usada para resolver problemas de classificação binária. É uma das primeiras arquiteturas de redes neurais desenvolvida e tem sido amplamente utilizada como uma ferramenta básica para entender como as redes neurais aprendem.
- O processo de treinamento de um perceptron é baseado no algoritmo de aprendizado de perceptron, que é um algoritmo de otimização supervisionado. Ele é usado para ajustar os pesos dos neurônios para minimizar a diferença entre a saída desejada e a saída produzida pela rede. O algoritmo de aprendizado de perceptron é iterativo e é executado até que a diferença entre a saída desejada e a saída produzida pela rede seja menor que um determinado limite.

# Treinamento de um perceptron

O processo de treinamento de um perceptron consiste em três etapas:

1. Inicialização: os pesos dos neurônios são inicializados com valores aleatórios.
  2. Propagação: os dados de entrada são passados pela rede e a saída é produzida.
  3. Atualização: os pesos dos neurônios são atualizados com base na diferença entre a saída desejada e a saída produzida pela rede.
- A atualização dos pesos é realizada usando o algoritmo de aprendizado de perceptron, que é baseado no gradiente descendente. Ele usa a derivada dos erros em relação aos pesos para calcular a direção de atualização dos pesos. A taxa de aprendizado é um parâmetro que controla a velocidade de atualização dos pesos.
    - É importante notar que o perceptron só é capaz de resolver problemas linearmente separáveis, isto é, problemas em que os dados de treinamento podem ser separados por uma linha reta. Se os dados não são linearmente separáveis, o perceptron não será capaz de aprender uma solução correta.

# Backpropagation

- Backpropagation é um algoritmo de aprendizado de máquina utilizado para ajustar os pesos de uma rede neural artificial. Ele é usado para treinar redes neurais feedforward, que têm camadas sucessivas de neurônios conectados entre si. O algoritmo é chamado de backpropagation porque ele "propaga o erro de volta" pela rede, ajustando os pesos de cada camada para minimizar o erro.
- O backpropagation é um algoritmo iterativo, ou seja, o processo de treinamento é repetido várias vezes com diferentes conjuntos de dados até que o erro seja minimizado. A taxa de aprendizado é um parâmetro que controla a velocidade de atualização dos pesos.
  - O backpropagation é um algoritmo eficiente e robusto para treinar redes neurais feedforward, é utilizado em uma variedade de aplicações, como reconhecimento de imagens, processamento de linguagem natural, reconhecimento de fala e tomada de decisão.

# Backpropagation

O processo de treinamento com backpropagation consiste em três etapas principais:

1. Propagação para frente: os dados de entrada são passados pela rede e a saída é produzida. Isso é realizado pela aplicação das funções de ativação e dos pesos de cada neurônio na camada.
2. Cálculo do erro: a diferença entre a saída desejada e a saída produzida pela rede é calculada e usada como medida do erro.
3. Propagação para trás: o erro é propagado de volta pela rede, ajustando os pesos de cada neurônio para minimizar o erro. Isso é feito pelo cálculo dos gradientes dos erros em relação aos pesos usando o método do gradiente descendente.

# Treinamento do MLP

- MLP (*Multi-Layer Perceptron*) é uma arquitetura de rede neural artificial composta por várias camadas de neurônios, em que cada camada é composta por vários neurônios (como um cérebro humano). É uma extensão do perceptron simples, que é composta por uma única camada de neurônios. As camadas adicionais permitem que a rede realize operações mais complexas, o que permite resolver problemas não linearmente separáveis.
- O processo de treinamento de uma MLP é semelhante ao do perceptron, usando o algoritmo de backpropagation. Ele é um algoritmo de otimização supervisionado que ajusta os pesos dos neurônios para minimizar a diferença entre a saída desejada e a saída produzida pela rede. O algoritmo de backpropagation é iterativo e é executado até que a diferença entre a saída desejada e a saída produzida pela rede seja menor que um determinado limite.



# Treinamento do MLP

O processo de treinamento de uma MLP consiste em três etapas:

1. Inicialização: os pesos dos neurônios são inicializados com valores aleatórios.
2. Propagação para frente: os dados de entrada são passados pela rede e a saída é produzida. Isso é realizado pela aplicação das funções de ativação e dos pesos de cada neurônio em cada camada.
3. Propagação para trás: o erro é propagado de volta pela rede, ajustando os pesos de cada neurônio para minimizar o erro. Isso é feito pelo cálculo dos gradientes dos erros em relação aos pesos usando o método do gradiente descendente.

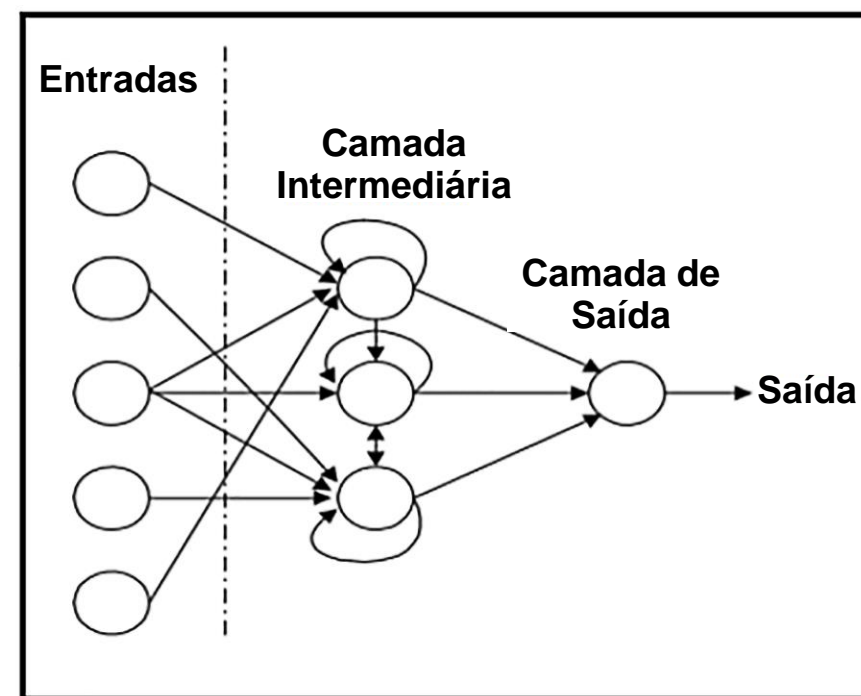
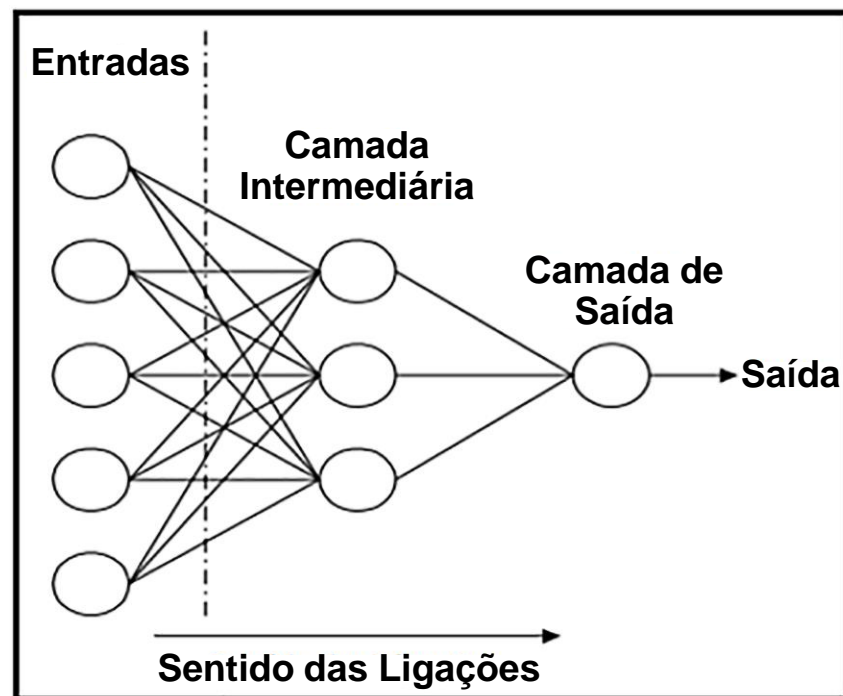
# Treinamento do MLP

- A diferença do perceptron para o MLP é que a MLP possui mais camadas e assim tem mais capacidade de representação e pode realizar tarefas mais complexas. A taxa de aprendizado é um parâmetro que controla a velocidade de atualização dos pesos e é importante para evitar o *overfitting*.
- É importante notar que o treinamento de uma MLP pode ser computacionalmente caro, dependendo do tamanho da rede e do tamanho do conjunto de dados de treinamento. Algumas técnicas, como o uso de técnicas de otimização e regularização, podem ser usadas para melhorar o desempenho e a eficiência do treinamento.
  - A camada de entrada não realiza processamento, apenas distribui os valores de entrada para os neurônios de primeira camada de processamento. Podemos conectar os neurônios artificiais de duas maneiras: com ou sem realimentação, podendo assim dividir as MLP em duas classes (BRAGA; LUDERMIR; CARVALHO, 2000).

# Treinamento do MLP

- À direita, uma RNA não recorrente e à esquerda uma RNA recorrente.
- MLP não recorrente (*Feed-Forward*): as saídas não realimentam as entradas, os neurônios da mesma camada e nem os neurônios das camadas anteriores são conectados. As entradas são aplicadas apenas aos neurônios da primeira camada.
- MLP recorrente: devido à realimentação das saídas para as entradas, as saídas dependem das entradas atuais e das saídas anteriores, pode haver conexões de neurônios de uma mesma camada.

Fonte: adaptado de: Braga; Ludermir; Carvalho (2000).



# Por que utilizar redes neurais?

Algumas das principais razões para utilizar redes neurais incluem:

- Capacidade de aprendizado: as redes neurais são capazes de aprender e se adaptar a novos dados, o que as torna úteis para tarefas que exigem aprendizado automático. Por exemplo, elas podem ser usadas para reconhecimento de imagens, processamento de linguagem natural e tomada de decisão.
- Robustez: as redes neurais são robustas a ruído e variações na entrada, o que as torna úteis para tarefas que enfrentam incerteza e ambiguidade.
- Capacidade de generalização: as redes neurais são capazes de generalizar suas habilidades para novos dados, o que as torna úteis para tarefas que exigem generalização para novos exemplos.

# Por que utilizar redes neurais?

- Capacidade de representação: as redes neurais são capazes de representar relações complexas entre entradas e saídas, o que as torna úteis para tarefas que exigem representação de conhecimento.
- Flexibilidade: as redes neurais podem ser usadas para resolver uma ampla variedade de tarefas, desde problemas simples até problemas mais complexos, e podem ser facilmente adaptadas para novas tarefas.
- Escalabilidade: as redes neurais podem ser escaladas para lidar com grandes conjuntos de dados, o que as torna úteis para tarefas que exigem grande capacidade de processamento.

# Aplicações de redes neurais artificiais

Algumas das principais aplicações de redes neurais artificiais incluem:

- Visão computacional: as redes neurais são utilizadas em aplicações de visão computacional, como reconhecimento de imagens e detecção de objetos. Elas são capazes de reconhecer rostos, reconhecer objetos em imagens, rastrear objetos em vídeos e muito mais.
- Processamento de linguagem natural: as redes neurais são usadas em tarefas de processamento de linguagem natural, como tradução automática, gerador de texto, reconhecimento de voz e compreensão de linguagem.
  - Reconhecimento de fala: as redes neurais são usadas para transformar fala em texto e para reconhecer comandos de voz, como reconhecimento de voz para dispositivos de assistente de voz.

# Aplicações de redes neurais artificiais

- Robótica: as redes neurais são usadas para controlar robôs e para permitir que os robôs aprendam e se adaptem a novas tarefas.
- Análise de dados: as redes neurais são usadas para analisar grandes conjuntos de dados, como dados de mercado financeiro, dados médicos e dados climáticos, para prever tendências e tomar decisões informadas.
- Jogos: as redes neurais são usadas para desenvolver jogos inteligentes, como jogos de tabuleiro e jogos de videogame, em que os jogadores podem jogar contra agentes de inteligência artificial.

# Interatividade

Analise as afirmações:

- I. O perceptron é uma arquitetura simples de rede neural artificial, composta por uma única camada de neurônios.
- II. O perceptron só é capaz de resolver problemas não linearmente separáveis.
- III. Backpropagation é um algoritmo de aprendizado de máquina utilizado para ajustar os pesos de uma rede neural artificial.
- IV. Uma rede neural perceptron multicamada é uma arquitetura de rede neural artificial composta por várias camadas de neurônios.
- V. As camadas adicionais em uma rede neural permitem resolver problemas linearmente separáveis.

Estão corretas:

- a) I, II e III.
- b) I, III e IV.
- c) II, III e V.
- d) III, IV e V.
- e) Todas as afirmações.



# Resposta

Analise as afirmações:

- I. O perceptron é uma arquitetura simples de rede neural artificial, composta por uma única camada de neurônios.
- II. O perceptron só é capaz de resolver problemas não linearmente separáveis.
- III. Backpropagation é um algoritmo de aprendizado de máquina utilizado para ajustar os pesos de uma rede neural artificial.
- IV. Uma rede neural perceptron multicamada é uma arquitetura de rede neural artificial composta por várias camadas de neurônios.
- V. As camadas adicionais em uma rede neural permitem resolver problemas linearmente separáveis.

Estão corretas:

- a) I, II e III.
- b) I, III e IV.**
- c) II, III e V.
- d) III, IV e V.
- e) Todas as afirmações.

# Algoritmos genéticos

- Os algoritmos genéticos (AGs) são uma técnica de inteligência artificial baseada no processo de evolução natural. Eles são utilizados para resolver problemas de otimização e busca de soluções em áreas como engenharia, ciência da computação e negócios.
- Os AGs usam conceitos da biologia evolutiva, como seleção natural, reprodução e mutação, para encontrar soluções para problemas complexos. Eles criam uma população de soluções candidatas (chamadas de indivíduos) e, por meio de uma série de iterações, selecionam os indivíduos mais aptos para reprodução e combinação, enquanto outros são descartados. A cada iteração, a população evolui para se aproximar de uma solução ótima para o problema.
  - Os AGs são utilizados para resolver problemas de otimização, como encontrar os melhores parâmetros para um modelo, encontrar a melhor rota em um grafo ou encontrar a melhor estratégia em um jogo. Também são utilizados para resolver problemas de classificação e previsão, como classificar documentos ou prever preços de ações.

# Algoritmos genéticos

- Os AGs são capazes de lidar com problemas complexos e não lineares, e podem encontrar soluções ótimas mesmo quando não há uma solução analítica disponível. Eles também são flexíveis e podem ser facilmente adaptados para novos problemas. No entanto, os AGs podem ser computacionalmente caros e podem não ser a melhor escolha para problemas simples ou para problemas com restrições de tempo.

Existem vários tipos de algoritmos genéticos (AGs), cada um com suas próprias características e aplicações específicas. Alguns dos principais tipos de AGs incluem:

- Algoritmo Genético Clássico (AGC): é um dos tipos mais comuns de AGs e é baseado nas ideias básicas da evolução natural, como seleção, reprodução e mutação. Ele é utilizado para resolver problemas de otimização e busca de soluções.
- Algoritmo Genético Evolutivo (AGE): é uma variação do AGC que usa técnicas evolutivas adicionais, como o uso de diferentes taxas de mutação e cruzamento, a fim de aumentar a eficiência do algoritmo.

# Algoritmos genéticos

- Algoritmo Genético Estocástico (AGE): é uma variação do AGC que utiliza métodos estocásticos para seleção, reprodução e mutação, a fim de evitar o problema de convergência precoce.
- Algoritmo Genético de Mapa de Otimização (AGMO): é um AG baseado no mapa de otimização e é utilizado para resolver problemas de otimização não linear.
- Algoritmo Genético de Programação (AGP): é um AG baseado na programação genética e é utilizado para resolver problemas de programação.
- Algoritmo Genético Multiobjetivo (AGMO): é um AG projetado para resolver problemas de otimização multiobjetivo, em que mais de uma função deve ser otimizada simultaneamente.
  - Algoritmo Genético de Partícula (AGP): é um AG baseado no algoritmo de otimização de enxame de partículas (PSO) e é utilizado para resolver problemas de otimização não linear.

# Conceitos básicos

Os conceitos básicos de algoritmos genéticos incluem:

- **População:** a população é composta por um conjunto de soluções candidatas (chamadas de indivíduos) para o problema. Cada indivíduo é representado por um conjunto de características ou parâmetros, chamados de cromossomos.
- ***Fitness*:** a *fitness* é uma medida da aptidão de um indivíduo para resolver o problema. Ela é geralmente calculada com base em uma função de *fitness*, que avalia como bem um indivíduo se adapta às restrições e objetivos do problema.
  - **Seleção:** a seleção é o processo de escolha dos indivíduos mais aptos para reprodução. Ela geralmente é baseada em sua aptidão, mas também pode considerar outros critérios, como diversidade genética.

# Conceitos básicos

- Cruzamento (*crossover*): o cruzamento é o processo de combinar os cromossomos de dois indivíduos para produzir novos indivíduos. Ele é geralmente realizado escolhendo aleatoriamente pontos de corte nos cromossomos dos pais e combinando as partes correspondentes dos cromossomos para produzir os cromossomos dos filhos.
- Mutação: a mutação é o processo de alterar aleatoriamente um ou mais cromossomos de um indivíduo. Isso é feito para introduzir nova diversidade genética na população e evitar a convergência precoce.
  - Iteração: o processo de seleção, cruzamento e mutação é repetido em cada iteração, produzindo uma nova geração de indivíduos.

# Funcionamento do algoritmo

O funcionamento básico de um AG é o seguinte:

- Inicialização: o algoritmo começa criando uma população inicial de indivíduos aleatórios, em que cada indivíduo é representado por um conjunto de características ou parâmetros, chamados de cromossomos.
- Avaliação: cada indivíduo é avaliado usando uma função de *fitness*, que mede sua aptidão para resolver o problema. A função de *fitness* é geralmente definida com base nas restrições e objetivos do problema.
  - Seleção: a seleção é o processo de escolher os indivíduos mais aptos para reprodução. Ela geralmente é baseada na aptidão dos indivíduos, mas também pode considerar outros critérios, como diversidade genética.

# Funcionamento do algoritmo

- Cruzamento (*crossover*): o cruzamento é o processo de combinar os cromossomos de dois indivíduos para produzir novos indivíduos.
- Mutação: a mutação é o processo de alterar aleatoriamente um ou mais cromossomos de um indivíduo.
- Iteração: o processo de seleção, cruzamento e mutação é repetido em cada iteração, produzindo uma nova geração de indivíduos.
- Os AGs são capazes de lidar com problemas complexos e não lineares, e podem encontrar soluções ótimas mesmo quando não há uma.



# Seleção dos mais aptos

- A seleção é um dos passos fundamentais do processo de algoritmo genético (AG). Ela é responsável por escolher os indivíduos mais aptos para reprodução e, conseqüentemente, para a formação de uma nova geração. O objetivo da seleção é garantir que os indivíduos mais aptos sejam reproduzidos com maior frequência, a fim de aumentar as chances de obter soluções ótimas para o problema.
- A escolha do método de seleção a ser utilizado depende do problema específico e do contexto em que o AG está sendo aplicado.

# Seleção dos mais aptos

Existem vários métodos de seleção utilizados em AGs, cada um com suas próprias vantagens e desvantagens. Alguns dos métodos mais comuns incluem:

- Seleção por roleta: esse método é baseado na ideia de que indivíduos com maior aptidão têm maiores chances de serem selecionados para reprodução. Ele funciona atribuindo uma porcentagem de escolha para cada indivíduo, proporcional à sua aptidão.
- Seleção por torneio: esse método envolve escolher aleatoriamente um grupo de indivíduos e selecionar o indivíduo com a maior aptidão para reprodução. Isso é repetido várias vezes para formar uma nova geração.
  - Seleção estocástica universal: esse método se baseia no princípio de seleção proporcional à aptidão, mas utiliza uma técnica estocástica para escolher indivíduos, evitando que indivíduos com maior aptidão sejam selecionados com muita frequência.
  - Seleção por elitismo: esse método consiste em preservar os indivíduos mais aptos de uma geração para a próxima, garantindo que as características desejadas não sejam perdidas.

# Parâmetros genéticos

Os parâmetros genéticos são uma parte importante do algoritmo genético (AG) e são usados para controlar o comportamento do algoritmo. Eles incluem a taxa de *crossover*, taxa de mutação, tamanho da população, número de gerações, entre outros. A seguir, são descritos alguns dos principais parâmetros genéticos:

- Taxa de *crossover*: é a probabilidade de que um par de indivíduos sofra *crossover* (ou seja, troca de informação genética) durante a reprodução. É geralmente definida como um valor entre 0 e 1. Valores mais altos de taxa de *crossover* geralmente resultam em uma maior diversidade genética, mas também aumentam a chance de introduzir características indesejadas na população.
  - Taxa de mutação: é a probabilidade de que um cromossomo sofra mutação (ou seja, alteração aleatória) em cada geração. É geralmente definida como um valor entre 0 e 1. Valores mais altos de taxa de mutação geralmente resultam em uma maior diversidade genética, mas também aumentam a chance de introduzir características indesejadas na população.

# Parâmetros genéticos

- Tamanho da população: esse parâmetro controla o número de indivíduos presentes na população. Um tamanho de população maior pode aumentar a diversidade genética, mas também pode aumentar o tempo de processamento.
- Taxa de cruzamento: esse parâmetro controla a frequência com que ocorre o cruzamento entre indivíduos. Uma taxa de cruzamento alta pode aumentar a diversidade genética, mas também pode aumentar o risco de perda de informação útil.
  - Número de gerações: esse parâmetro refere-se ao número de iterações que o algoritmo executa durante o processo de otimização. Cada geração é composta por um conjunto de indivíduos (soluções candidatas) que são avaliados de acordo com uma função de aptidão, que determina quão bem cada indivíduo resolve o problema em questão.

# Aplicações

Algoritmos genéticos são aplicados em uma variedade de campos, incluindo:

- Engenharia: para otimizar o desempenho de sistemas mecânicos, elétricos e de controle. Por exemplo, eles são utilizados para projetar antenas, otimizar a configuração de sistemas de controle de processos industriais e encontrar soluções para problemas de planejamento de projetos.
- Finanças: para otimizar a alocação de ativos, encontrar estratégias de negociação e prever o desempenho de ações.
- Ciência da computação: para resolver problemas de otimização em inteligência artificial, como redes neurais, algoritmos de aprendizado de máquina e algoritmos de busca.
  - Bioinformática: para analisar dados de sequenciamento de DNA e proteínas, identificar genes relacionados a doenças e desenvolver novos medicamentos.
  - Robótica: para otimizar a performance de robôs, desenvolver estratégias de navegação e planejar trajetórias.

# Interatividade

Analise as afirmações:

- I. Os algoritmos genéticos são uma técnica de inteligência artificial baseada no processo de evolução natural.
- II. Os algoritmos genéticos são capazes de lidar com problemas complexos e não lineares.
- III. O cruzamento é o processo de combinar os cromossomos de dois indivíduos para produzir novos indivíduos.
- IV. A mutação é o processo de alterar aleatoriamente um ou mais cromossomos de um indivíduo.
- V. A seleção é o processo de escolher os indivíduos mais aptos para reprodução.

Estão corretas:

- a) I, II e III.
- b) I, II e IV.
- c) I, II e V.
- d) III, IV e V.
- e) Todas as afirmações.

# Resposta

Analise as afirmações:

- I. Os algoritmos genéticos são uma técnica de inteligência artificial baseada no processo de evolução natural.
- II. Os algoritmos genéticos são capazes de lidar com problemas complexos e não lineares.
- III. O cruzamento é o processo de combinar os cromossomos de dois indivíduos para produzir novos indivíduos.
- IV. A mutação é o processo de alterar aleatoriamente um ou mais cromossomos de um indivíduo.
- V. A seleção é o processo de escolher os indivíduos mais aptos para reprodução.

Estão corretas:

- a) I, II e III.
- b) I, II e IV.
- c) I, II e V.
- d) III, IV e V.
- e) Todas as afirmações.

# Referências

- BISHOP, C. *Pattern Recognition and Machine Learning* (Information Science and Statistics). Secaucus: Springer-Verlag New York, Inc., 2006.
- BRAGA, A.; LUDERMIR, T.; CARVALHO, A. *Redes neurais artificiais: teoria e aplicações*. Rio de Janeiro: LTC, 2000.
- DUDA, R.; HART, P.; STORK, D. *Pattern classification*. 2nd edition. New York: John Wiley Sons, Inc., 2000.
- HAYKIN, S. *Neural networks: a comprehensive foundation*. New York: MacMillan, 1994.
- NORVIG, P. *Inteligência Artificial*. 3. ed. São Paulo: Grupo GEN, 2013.



**ATÉ A PRÓXIMA!**