

# Residência brisa

### Banco de dados

## 1. Introdução a banco de dados

### 1.1 Introdução a Banco de Dados: Um Resumo Simplificado

- Conceitual:

Na fase conceitual, utilizamos o Diagrama de Entidade-Relacionamento (DER) para representar abstratamente o banco de dados discutido com o cliente. No DER, identificamos as entidades, que podem ser físicas, lógicas, fortes ou fracas. Cada ocorrência de entidades é adicionada como uma entrada na tabela. Os atributos são características das entidades, podendo ser simples, compostos, monovalorados, multivalorados, armazenados ou derivados. Os relacionamentos representam as ações ocorridas entre as entidades, como pegar, perder e vender. A cardinalidade define o número de entidades que podem se relacionar, como um para muitos (1..n ou 1..\*) ou um para um (1..1). Também existem relacionamentos muitos para muitos. Em casos específicos, podemos ter relacionamentos ternários. A especialização permite que uma entidade herde características de outra, como um funcionário que se especializa em engenharia.

- Lógico:

No modelo lógico, podemos optar por um modelo relacional ou não relacional. O modelo relacional é baseado em conceitos matemáticos de conjuntos e é estruturado em tabelas. Os atributos continuam representando características e os valores são preenchidos nas células de cada atributo. As linhas das tabelas são chamadas de tuplas e as tabelas em si são representações dos dados.

- Físico:

Na fase física, utilizamos um gerenciador de banco de dados, como o MySQL, e uma linguagem de consulta chamada SQL. O phpMyAdmin é uma ferramenta comumente usada para manipular tabelas e consultar estruturas de banco de dados. Nesse estágio, é necessário entender a estrutura do banco de dados e como executar comandos para modificá-lo.

- Modelos de Bancos de Dados:

Existem diferentes modelos de bancos de dados, sendo os mais comuns o modelo relacional e o orientado a objetos. No modelo relacional, as linhas das tabelas são chamadas de tuplas e as colunas são atributos. É importante que os valores dos atributos sejam armazenados da mesma forma para todas as tuplas. Cada tabela precisa de um atributo identificador, geralmente chamado de ID. Além disso, temos conceitos como domínio (descrição física e semântica dos valores permitidos), grau de relação (quantidade de atributos na tabela), esquema (padrão de organização das tabelas), chave primária (única, não nula e não redundante) e chave estrangeira (chave que relaciona uma tabela a outra).

O modelo orientado a objetos é mais direcionado para aplicações de negócios e suporta consultas mais elaboradas. Ele é comumente usado em Data Warehouses, que são bancos de dados especializados em análises de negócios. A modelagem dimensional é baseada em métricas que descrevem aspectos do negócio. Os fatos representam itens, transações ou eventos, enquanto as dimensões permitem diferentes visualizações dos fatos (por ano, mês, vendedor, etc.). As métricas são atributos numéricos que representam os fatos, como o número de

vendas. Essas informações podem ser representadas em um cubo de várias dimensões. Para análises mais complexas, são utilizadas ferramentas como OLAP (Online Analytical Processing) e outras ferramentas de Business Intelligence (BI).

### ### 1.2 Banco de dados não relacional SQL

#### Introdução a Bancos de Dados Não Relacionais: Um Resumo Simplificado

- Dados não são em tabelas:

Nos bancos de dados não relacionais, os dados não são organizados em tabelas como no modelo relacional. Eles oferecem alta escalabilidade, permitindo que os dados sejam particionados em vários servidores para melhor desempenho. Além disso, esses bancos são flexíveis em relação aos esquemas, sendo utilizados modelos como orientado a documentos, chave/valor e orientado a grafos.

- Orientado a documentos:

Nesse modelo, os dados são armazenados em documentos, geralmente no formato JSON ou XML. Diferentemente do modelo relacional, os documentos não precisam ter uma estrutura fixa, o que oferece maior flexibilidade. Os documentos são guardados no banco de dados e podem ser consultados de forma eficiente.

- Chave/valor:

Esse modelo permite a associação de valores a chaves, possibilitando a recuperação rápida dos dados apenas através das chaves. É uma estrutura simples em que cada valor é armazenado e recuperado por uma chave correspondente.

- Orientado a grafos:

No modelo de banco de dados orientado a grafos, os dados são representados como nós e arestas, permitindo a criação de relacionamentos complexos. Existem diferentes tipos de grafos, como grafos simples, hipergrafos, grafos com atributos e grafos aninhados. Os nós podem ter rótulos (labels) ou atributos (chave/valor), e as arestas podem ter rótulos ou atributos para melhorar os relacionamentos. Esse modelo é especialmente útil quando há necessidade de representar e analisar relações complexas entre os dados.

- Orientado a colunas:

Esse modelo organiza os dados em tuplas, mas com uma abordagem mais distante do modelo relacional tradicional. Os dados são armazenados em colunas, o que permite uma busca escalável e oferece flexibilidade para adicionar colunas posteriormente. Além disso, os bancos de dados orientados a colunas são capazes de comprimir os dados, resultando em melhor desempenho e economia de espaço.

- Complexidade dos bancos:

Os diferentes modelos de bancos de dados não relacionais possuem diferentes níveis de complexidade. O modelo chave/valor é o mais simples, seguido pelo modelo orientado a colunas. O modelo orientado a documentos apresenta uma complexidade maior, mas sem a complexidade dos relacionamentos. Já o modelo orientado a grafos possui uma alta complexidade em relação aos relacionamentos, porém uma baixa complexidade em relação aos dados em si.

- Melhores SGBDs gerenciadores de dados:

Existem diferentes bancos de dados não relacionais disponíveis, cada um com suas características e casos de uso específicos. Alguns dos principais são:

1. **DynamoDB:** Um banco de dados chave/valor que utiliza tabelas para organizar os dados. Os itens representam as ocorrências e os atributos são as características dessas ocorrências. Possui suporte para atributos aninhados e é altamente escalável.
2. **MongoDB:** É um banco de dados orientado a documentos que não requer esquemas fixos. Os dados são armazenados em documentos individuais, sendo especialmente adequado para uso em Big Data e flexibilidade de estrutura.
3. **ArangoDB:** Um banco de dados que suporta modelos chave/valor, documentos e grafos. Possui uma linguagem de consulta unificada chamada ArangoDB Query Language (AQL) e permite a criação de relacionamentos por meio de documentos de vértices e arestas.
4. **Cassandra Database:** Um banco de dados orientado a colunas com foco em armazenamento barato e alta escalabilidade. Utiliza a linguagem CQL para manipulação e definição dos bancos, permitindo a criação de keyspaces, tabelas e operações de consulta, inserção, atualização e exclusão de dados.

Existem também outros bancos de dados NoSQL disponíveis no mercado, cada um com suas vantagens e casos de uso específicos. A escolha do SGBD depende das necessidades do projeto e dos requisitos de escalabilidade, desempenho e flexibilidade.

### ### 1.3 Banco de Dados NewSQL, Big Data, Internet das Coisas e Computação em Nuvem

Resumo sobre Banco de Dados NewSQL, Big Data, Internet das Coisas e Computação em Nuvem:

## Banco de Dados NewSQL:

- Os bancos de dados NewSQL são considerados novas gerações de bancos de dados, que combinam características de desempenho e consistência dos bancos de dados relacionais tradicionais com escalabilidade e distribuição dos bancos de dados NoSQL.
- Esses bancos de dados são projetados para lidar com grandes volumes de dados e um alto número de requisições, especialmente em cenários que exigem alta taxa de performance e consistência.
- Eles utilizam a linguagem SQL (Structured Query Language) para consulta e manipulação de dados, além de suportarem transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade) para garantir a integridade dos dados.
- Os bancos de dados NewSQL também oferecem controle de concorrência não excludente, o que permite a realização de várias operações simultâneas sem prejudicar a consistência dos dados.
- Eles são projetados para arquiteturas distribuídas, o que significa que podem ser escalados e distribuídos em vários servidores para melhorar o desempenho e a disponibilidade.

## Diferenças entre NewSQL e NoSQL:

- Enquanto os bancos de dados NewSQL suportam transações ACID, os bancos de dados NoSQL geralmente não fornecem garantias de consistência transacional.
- Os bancos de dados NewSQL são adequados para aplicativos que exigem processamento de transações online (OLTP) e também análise de dados em tempo real (OLAP) em data warehouses. Os bancos de dados NoSQL também podem lidar com esses tipos de aplicativos, mas sua complexidade aumenta consideravelmente.
- Os bancos de dados NewSQL têm menor flexibilidade em relação a tipos e formatos de dados, enquanto os bancos de dados NoSQL são altamente flexíveis e podem armazenar dados de diferentes estruturas.
- Os bancos de dados NewSQL são conhecidos por sua alta velocidade e oferecem uma sobrecarga de desempenho mínima, enquanto os bancos de dados NoSQL têm um bom desempenho, mas podem ter uma sobrecarga moderada em certas situações.

## Big Data:

- Big Data refere-se à obtenção, análise e tratamento de grandes volumes de dados provenientes de diversas fontes.
- É caracterizado pelos "5Vs": Volume (grande quantidade de dados), Velocidade (alta velocidade de geração e processamento), Variedade (diversidade de fontes e formatos de

dados), Veracidade (verificação da veracidade dos dados) e Valor (extração de insights e tomada de decisões com base nos dados).

- Os bancos de dados NewSQL são adequados para lidar com Big Data, pois oferecem alta escalabilidade, desempenho e capacidade de processar grandes volumes de dados em tempo real.

Internet das Coisas (IoT):

- A IoT se refere à conexão de objetos físicos à rede, permitindo a comunicação e troca de informações entre eles.

- Essa tecnologia tem aplicação em diversos setores, como saúde, transporte, agricultura, automação residencial, entre outros.

- Os dados gerados pela IoT são tipicamente grandes em volume, variados em formatos e requerem processamento em tempo real.

- Os bancos de dados NewSQL são adequados para

lidar com os desafios da IoT, pois oferecem escalabilidade, desempenho e capacidade de processar grandes volumes de dados gerados por dispositivos conectados.

Computação em Nuvem e Banco de Dados:

- A computação em nuvem envolve a disponibilização de serviços de computação, como servidores, bancos de dados e software, por meio da internet.

- Existem três tipos de nuvem: pública, privada e híbrida.

- Os bancos de dados NewSQL se beneficiam da computação em nuvem, pois permitem que os usuários não se preocupem com infraestrutura, atualizações automáticas, compartilhamento de dados, acesso remoto e controle de custos.

- Esses bancos de dados ajudam a superar as limitações de desempenho associadas à computação em nuvem, oferecendo alta velocidade de processamento e capacidade de escala.

### 1.4 Linguagens SQL, NoSQL e NewSQL.

Resumo sobre Linguagens SQL, NoSQL e NewSQL:

Linguagem SQL:

- SQL (Structured Query Language) é uma linguagem usada para definir e manipular bancos de dados relacionais.

- É dividida em subconjuntos principais: DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), DTL (Data Transaction Language) e CQL (Consulta SQL).

- DDL é usado para criar, modificar e excluir objetos do banco de dados, como tabelas.

- DML é responsável por inserir, atualizar e excluir dados nas tabelas.

- DCL é usado para conceder ou remover acesso de usuários ao banco de dados.

- DTL é usado para controlar transações de dados, como commit e rollback.

- CQL (Consulta SQL) é o subconjunto mais usado e inclui principalmente o comando SELECT para realizar consultas e obter resultados.

phpMyAdmin:

- O phpMyAdmin é uma ferramenta de administração de banco de dados MySQL baseada na web.

- É usado para facilitar tarefas como criação de tabelas, inserção de dados, consulta e gerenciamento de usuários.

Etapas da Modelagem de Banco de Dados:

- A modelagem de banco de dados envolve a criação de uma representação estruturada do banco de dados.

- As etapas principais são: modelo conceitual, modelagem lógica e modelagem física.

- O modelo conceitual descreve as entidades, atributos e relacionamentos de alto nível do banco de dados.

- A modelagem lógica envolve a tradução do modelo conceitual em um modelo específico de banco de dados, como o modelo relacional.

- A modelagem física trata da implementação física do banco de dados, considerando aspectos como índices, particionamento e otimização de desempenho.

DDL - Linguagem de Definição de Dados:

- DDL inclui comandos como CREATE, DROP e ALTER.

- CREATE é usado para criar objetos, como tabelas.
- DROP é usado para excluir objetos, como tabelas ou bancos de dados.
- ALTER é usado para modificar objetos existentes, como adicionar ou remover colunas de uma tabela.

#### DML - Linguagem de Manipulação de Dados:

- DML é usado para inserir, atualizar e excluir dados em tabelas.
- Os principais comandos são INSERT, UPDATE e DELETE.
- INSERT é usado para inserir novos registros em uma tabela.
- UPDATE é usado para atualizar registros existentes em uma tabela.
- DELETE é usado para excluir registros de uma tabela.

#### DQL - Linguagem de Consulta de Dados:

- DQL é a parte mais importante da linguagem SQL, pois é usada para consultas e obtenção de resultados.
- O principal comando é o SELECT, usado para especificar quais campos devem ser retornados e quais condições devem ser aplicadas para filtrar os resultados.
- Cláusulas como FROM, WHERE, GROUP BY e ORDER BY são usadas para definir a origem dos dados, aplicar filtros, agrupar resultados e ordenar os resultados, respectivamente.
- Funções como AVG, COUNT, MIN, MAX e SUM podem ser usadas para realizar cálculos nos resultados das consultas.
- Operadores relacionais como >, <, >=, <=, = e <> são usados para comparar valores.

#### Banco de Dados NoSQL:

- O NoSQL é

uma abordagem de banco de dados que difere dos bancos de dados relacionais tradicionais.

- MongoDB é um exemplo popular de banco de dados NoSQL orientado a documentos.

- No NoSQL, os dados são organizados em coleções, semelhantes a tabelas, e os documentos são compostos por pares de chave e valor.
- Os documentos em uma coleção podem ter estruturas diferentes, oferecendo flexibilidade em relação aos bancos de dados relacionais.

Banco de Dados NewSQL:

- O NewSQL é uma abordagem que combina a escalabilidade e o desempenho do NoSQL com a linguagem SQL tradicional.
- Os bancos de dados NewSQL permitem o uso de SQL para acessar e manipular dados distribuídos em um ambiente escalável.
- Eles oferecem a capacidade de processar grandes volumes de dados e lidar com os desafios da IoT e da computação em nuvem.

``sql

Exemplos de comandos SQL:

DDL (Data Definition Language):

- CREATE TABLE: cria uma tabela.

Exemplo: CREATE TABLE Clientes (

id INT PRIMARY KEY,

nome VARCHAR(50),

idade INT

);

- DROP TABLE: exclui uma tabela.

Exemplo: DROP TABLE Clientes;

- ALTER TABLE: altera uma tabela.

Exemplo: ALTER TABLE Clientes ADD email VARCHAR(50);

ALTER TABLE Clientes DROP COLUMN idade;

DML (Data Manipulation Language):

- INSERT INTO: insere dados em uma tabela.

Exemplo: INSERT INTO Clientes (id, nome, idade)

VALUES (1, 'João', 25);



- UPDATE: atualiza registros em uma tabela.

Exemplo: UPDATE Clientes

```
SET nome = 'Maria', idade = 30
```

```
WHERE id = 1;
```

- DELETE FROM: exclui registros de uma tabela.

Exemplo: DELETE FROM Clientes WHERE id = 1;

DQL (Data Query Language):

- SELECT: consulta dados de uma tabela.

Exemplo: SELECT \* FROM Clientes;

```
SELECT nome, idade FROM Clientes WHERE idade > 20;
```

```
SELECT COUNT(*) FROM Clientes;
```

Exemplos de comandos NoSQL (MongoDB):

Inserção de um documento em uma coleção:

```
- db.clientes.insertOne({  
  nome: 'João',  
  idade: 25,  
  email: 'joao@example.com'  
});
```

Consulta de dados em uma coleção:

```
- db.clientes.find({ nome: 'João' });  
- db.clientes.find({ idade: { $gt: 20 } });  
- db.clientes.find().sort({ idade: -1 }).limit(10);
```

Atualização de um documento em uma coleção:

```
- db.clientes.updateOne(  
  { nome: 'João' },  
  { $set: { idade: 30 } }  
);
```

Remoção de um documento em uma coleção:

```
- db.clientes.deleteOne({ nome: 'João' });
```

Esses são apenas exemplos básicos dos comandos SQL e NoSQL. Existem muitas outras possibilidades e recursos disponíveis em cada linguagem e banco de dados.

...

## ## 2. Introdução noções de Big Data.

### ### 2.1 Introdução a Big Data:

Noções de Big Data:

Big Data refere-se ao grande volume, velocidade e variedade de dados que são gerados diariamente. Esses dados podem ser estruturados, como informações armazenadas em bancos de dados relacionais, não estruturados, como texto, áudio e vídeo, ou semi-estruturados, que possuem algum tipo de organização, mas não seguem um formato rígido. O conceito de Big Data também engloba os desafios associados ao armazenamento, processamento e análise desses dados para extrair informações e insights valiosos.

Evolução da Big Data Science no Mundo:

A evolução da Big Data Science pode ser rastreada por meio de marcos importantes:

1. MapReduce: O Google desenvolveu a estrutura MapReduce, que permitia o processamento paralelo e distribuído de grandes volumes de dados em clusters de hardware simples.
2. Hadoop: Com base no artigo do Google sobre MapReduce, o Yahoo criou o Hadoop, uma plataforma de código aberto que permitia a execução de operações distribuídas em clusters de servidores, tornando a análise de Big Data mais acessível.
3. Adoção do Hadoop: Grandes empresas como Facebook e eBay adotaram o Hadoop para lidar com o processamento de grandes volumes de dados e a execução de tarefas complexas em escala.
4. Bancos de dados NoSQL: Com o aumento da necessidade de lidar com dados não estruturados, surgiu uma nova categoria de bancos de dados chamada NoSQL. Um exemplo é o Apache Spark, que se integrou ao ecossistema Hadoop e forneceu recursos avançados de processamento em tempo real.

5. Aumento do uso do código aberto: Com o advento de tecnologias de Big Data de código aberto, houve um aumento significativo na adoção dessas ferramentas pelas empresas. Isso resultou em um aumento no número de profissionais com habilidades em Big Data.

6. Desafios de segurança e eficiência: Embora tenham ocorrido avanços significativos, ainda há desafios relacionados à segurança e eficiência na implementação de algoritmos de Big Data. Além disso, a manipulação manual dos dados continua sendo uma tarefa importante em muitos casos.

#### Big Data Science no Brasil:

Embora o mercado de Big Data no Brasil esteja aquecido, ele é considerado relativamente pequeno em comparação com outros países. No entanto, as oportunidades estão crescendo à medida que as empresas reconhecem o valor dos dados e a necessidade de especialistas em Big Data para extrair informações valiosas e tomar decisões embasadas em dados.

#### Avanço da Big Data Science nos Postos de Trabalho:

Para evitar que seu trabalho seja substituído por automação ou robôs, é essencial desenvolver habilidades que vão além das tarefas mecânicas. Além das habilidades técnicas (hard skills), como conhecimento em programação, estatística e manipulação de dados, é importante desenvolver habilidades interpessoais (soft skills), como comunicação eficaz, pensamento crítico, trabalho em equipe e liderança.

#### Data Literacy:

Data Literacy refere-se à habilidade de compreender, analisar e comunicar efetivamente informações baseadas em dados. Em um mundo onde os dados estão se multiplicando rapidamente, a Data Literacy torna-se essencial para tomar decisões informadas e orientadas por dados. A capacidade de interpretar corretamente os dados, identificar padrões e extrair insights valiosos é fundamental para aproveitar o potencial da Big Data Science.

#### Principais Conceitos em Big Data Science:

- Matéria-prima para a Data Science: Informação, dado e conhecimento são os componentes fundamentais da ciência de dados. Os dados podem ser obtidos de várias fontes, como sensores, redes sociais, registros de transações, entre outros.

- Tipos de dados: Existem diferentes tipos de dados usados na análise de Big Data. Dados contínuos podem assumir qualquer valor em um intervalo, enquanto dados discretos são valores distintos sem aproximações. Dados categóricos representam um conjunto específico de categorias possíveis, como níveis de escolaridade. Dados binários são uma forma específica de dados categóricos, com apenas duas categorias possíveis, como 0 ou 1.

- Estatística vs. Ciência de Dados: A análise estatística é uma das etapas da ciência de dados, que envolve o ciclo completo de vida dos dados, desde a coleta até a interpretação dos resultados. A estatística se concentra na análise e inferência de dados, enquanto a ciência de dados vai além, incluindo a manipulação, processamento e visualização de dados.

Diferença entre Big Data, Data Science e Machine Learning:

- Big Data refere-se às características dos dados, como volume, velocidade e variedade.
- Data Science envolve o uso de técnicas e métodos para extrair informações valiosas dos dados, como análise exploratória, modelagem estatística e visualização de dados.
- Machine Learning é uma subárea da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos capazes de aprender e fazer previsões a partir dos dados, sem serem explicitamente programados.

Principais Instrumentos de Big Data Science:

Existem várias ferramentas utilizadas em Big Data Science, dependendo das necessidades e requisitos específicos de cada projeto. Alguns exemplos incluem Excel, PowerBI, RStudio, Python, Hadoop, Spark, entre outros. Essas ferramentas permitem a análise, manipulação e visualização de grandes volumes de dados, além de oferecerem recursos avançados para o processamento e modelagem de dados. A escolha das ferramentas adequadas depende do contexto e dos objetivos de cada projeto de Big Data Science.

### ## 3. Organização de computadores

#### ### Projetos De Circuitos Sequenciais3.1 Memórias Semicondutoras e Dispositivos Lógicos Programáveis: Resumo

Tipos de Memória:

Memórias de acesso aleatório (RAM) - Armazenamento randomico, volátil.

SRAM - utiliza latches, alta velocidade normalmente utilizadas em cache, também volátil.

DRAM - memória ram de acesso direto. Armazena dados em capacitores, faz uso de Refresh para manter os dados nos capacitores, exige circuito eletrônico adicional.

Memórias de somente Leitura

ROM - Apenas leitura dos dados, gravação de dados realizada pelo fabricante utiliza transistores MOSFET.

PROM - são produzidas pelo fabricante sem programação, usuário final programa de acordo com a sua aplicação, programação irreversível, já que precisa queimar resistores para ser 0 e deixá-los intactos para 1, utilizados em microcontroladores, dicionários eletrônicos, etc.

EPROM - Memória programável e apagável de somente leitura e EEPROM - Memória programável e apagável eletricamente somente leitura.

A EPROM é formada por transistores, para apagar as informações é através da injeção de radiação ultravioleta através de uma "janela" de quartzo transparente presente em seu invólucro.

Já memórias EEPROM pode ser programada através de pulsos elétricos. Podendo também ser apenas parcialmente apagada.

Memória flash - A flash é não volátil, escrita e leitura, grande capacidade de armazenamento, usada em pendrives.

Memória FIFO e LIFO - Memórias sequenciais.

FIFO - Conjunto de registradores de deslocamento que fazem operações de entrada e saída respeitando a condição: o primeiro dado escrito é o primeiro lido. possuindo assim prioridade de serem processados

LIFO - pilha de bytes operando de forma paralela. permite os dados serem armazenados e lidos na ordem inversa.

Tipos de armazenamento:

Discos rígidos - precisa ser hermeticamente conservado, feito normalmente de alumínio, vidro e/ou cerâmica com uma fina camada magnética.

Possuem eixos, braços e atuadores a sincronização das cabeças de escrita e leitura de dados, são escritos dos dois lados, tipicamente existem mais de um disco dispostos um sobre o outro, a velocidade de rotação pode variar são da ordem de 3600, 4500, 5400, 7200 ou 10.033 rpm

Armazenamento óptico - Utiliza lasers de baixa potência para escrita e leitura de dados, que produz e amplifica um feixe de luz, monocromático, coerente e colimado.

Exemplos: CD-ROM é um dispositivo de armazenamento óptico somente leitura. gravados pelo fabricante.

Basicamente é constituído por um plástico policarbonato, fina película metalizada de alumínio para leitura do laser infravermelho e uma cobertura base de verniz para proteção de oxidações.

Gravação realizada pelo fabricante na forma de entalhes (pits) sendo que a superfície plana ao redor é as lands.

O feixe infravermelho realiza a leitura em espiral e o CD-ROM gira sobre um eixo movido por motor 12v, os pits e lands representam os 0 e 1, e o laser os detecta e diferencia.

CD-R permite que o usuário grave diversos tipos de conteúdos. Ele utiliza o laser para queimar os pontos no plástico, que agora possui uma pintura orgânica para armazenar informação desejada. Como resultado a parte queimada difere a cor do restante da pintura, distinguindo 0 e 1, apenas uma gravação.

CD-RW - pode ser regravado várias vezes, para isso a superfície possui superfície de liga composta de prata, telúrio, índio e antimônio.

DVD-ROM - Disco de vídeo digital de somente leitura, grande capacidade de armazenamento chegando por ordem de 9GB, tamanho do pit bastante reduzido, permitindo maior armazenamento de dados.

BLU-RAY DISC: sendo capaz de armazenar filmes em alta qualidade com tempo de duração da ordem de 4h pode chegar até capacidade de 50GB.

#### Dispositivos Lógicos Programáveis

Para o desenvolvimento de Circuitos Integrados (CIs), o projetista desenvolve lógicas computacionais relativas às funções dos computadores. Em sistemas complexos gerando necessidade de grande espaço físico na placa de circuito impresso, por conta da necessidade de vários CIs. Foram desenvolvidas técnicas de integração para reduzir o tamanho dos CIs.

LSI - Integração em grande escala.

integração entre 1000 e 100000 componentes por pastilha. Normalmente utilizados para efetuar funções lógicas complexas como aritmética de calculadoras eletrônicas.

VLSI - Integração Elevada em Larga Escala.

integra entre 100000 e 10 milhões de componentes por pastilha, corresponde a manufatura de microprocessadores.

ULSI - Integração em Escala Ultra Larga.

integra 10 milhões pra cima de componentes por pastilha, utilizado em microprocessadores de sistemas computacionais avançados.

#### SPLD-PAL e SPLD-GAL

Um dispositivo lógico programável SPLD-PAL, consiste em um arranjo de portas lógicas AND e OR.

Possuem inúmeras entradas e saídas, baseiam-se na memória EPROM. Podem ser programados pelo usuário, servem para diversas utilizações práticas, microprocessadores e sistemas de controle processados. PAL é a marca registrada da AMD.

O SPLD-GAL é na verdade um SPLD-PAL que utiliza tecnologias para reprogramação como se fosse uma memória EEPROM. No lugar de fusíveis utiliza transistores.

#### CPLDs - Dispositivos Lógico Complexo Programável

Formado por diversos arranjos SPLDs. Existem vários LABs - Bloco de arranjo lógico com entradas e saídas de dados com interconexões programáveis por softwares conhecidos como PIA. Podem variar de acordo com a arquitetura utilizada, com a tecnologia empregada no processo de manufatura.

#### CPLD Altera MAX 7000

A Altera manufatura CPLDs de diversos tipos e séries incluindo a família MAX II, MAX 3000, MAX 7000 e MAX 9000. O CPLD MAX 7000, a sua estrutura contempla SPLD PAL e GAL, produzem funções de soma de produtos, podendo variar a densidade entre 2 e 16 LABs, utiliza memória semicondutora. Dependendo da série do CPLD, a densidade pode variar entre 2 a 16 LABs. Com relação à tecnologia de processo, o CPLD MAX 7000 utiliza a memória semicondutora EEPROM. A tensão de operação pode variar entre 3,3 e 5,0 V e o consumo de potência (energia) depende da tensão e da corrente elétrica, bem como da frequência elétrica de operação. Dessa forma, torna-se necessário consultar o datasheet (manual técnico de operação) do dispositivo lógico utilizado no projeto de sistemas digitais.

#### CPLD Xilinx CoolRunner II.

Existem diversas famílias e séries de CPLDs produzidos pela Xilinx como a família CoolRunner series XC2C32A XC2C64A.

A arquitetura do CoolRunner II baseia-se na estrutura de PLA ao invés da estrutura PAL, o PAL contempla um arranjo de portas lógicas AND programável, seguido por portas lógicas OR. Por outro lado o PLA utiliza arranjo de portas lógicas AND programável, seguido por um arranjo de portas lógicas OR também programáveis. O CPLD CoolRunner II contempla diversos blocos de funções FBs, Cada bloco contém 16 macrocélulas, exatamente como um LAB, conectados por uma matriz de interconexões (AIM) análoga às interconexões programáveis no MAX7000.

### ### 3.2 Projetos De Circuitos Sequenciais

#### Projetos de Circuitos Sequenciais

Primeiro é preciso distinguir um circuito combinacional de um circuito sequencial.

No circuito combinacional, os valores de saída dependem apenas dos valores de entrada em um determinado instante, pois não possuem memória. Quaisquer valores referentes à entrada e saídas que não estiverem inseridos no instante de processamento não têm efeito no circuito. Exemplos são codificadores, decodificadores, multiplexadores, demultiplexadores, circuitos aritméticos: somadores, meio somadores e subtratores.

No sequencial, os valores de saída em determinado instante não dependem apenas somente da sequência combinacional dos valores de entrada, mas sobretudo dos valores anteriores, ou seja, daqueles que a saída apresentava antes mesmo da atual sequência.

Tornando necessário elementos de memória como latches e os flip-flops. Computador é um circuito sequencial. Tanto as entradas como saídas do circuito sequencial estão conectadas ao circuito combinacional. Além disso, parte das saídas do circuito combinacional são as próprias entradas nos elementos de memória, sendo consideradas variáveis do próximo estado. Por sua vez, variáveis de saída dos elementos de memória constituem parte das entradas no circuito combinacional, sendo consideradas variáveis do estado atual. Sendo assim, um sistema realimentado, já que as variáveis de saída dos elementos de memória configuram-se como variáveis de entrada no circuito combinacional.

Latch - Dispositivos lógicos capazes de reter dois estados, SET E RESET. Biestáveis.

Semelhantes aos Flip-flops. No latch, a troca de estado ocorre de maneira assíncrona (depende da forma que as variáveis de entrada mudam o estado, não existe controle). Nos Flip-flops, de forma síncrona, controlados por um clock.

Latch RS - considerado elemento básico de memória ou memória simples, capacidade de armazenamento temporário de 1 bit, entrada Set configura a Saída Q para o nível lógico 1, e a entrada Reset limpa a saída Q para o nível lógico 0.

Sendo formado por duas portas lógicas NOR de acoplamento cruzado.

As saídas de cada porta lógica NOR, por sua vez, estão conectadas às entradas da porta lógica oposta, realimentação cruzada regenerativa (elos de realimentação).



Possível construir tabela-verdade.

Flip-flop - elemento de memória biestável síncrono, ou seja, a variável de saída muda conforme o clock. Função de busca e armazenamento de dados do estado anterior, para que este seja calculado e utilizado no próximo estado.

Existem quatro Flip-flops: Flip-flop RS (Reset-Set), Flip-flop JK, o Flip-flop T(toggle) e o Flip-flop D. Também conhecidos como multivibradores biestáveis, sendo encontrados nos computadores e na construção de registradores e contadores digitais.

#### Flip-flop RS

Quatro portas lógicas NAND. Duas fazem parte das entradas R e S com o clock e as outras estão conectadas com elos de realimentação nas saídas Q e !Q.

#### Flip-flop JK

Desenvolvido para corrigir o problema do estado ambíguo (R 1 e S 1) quando as entradas JK forem iguais ao nível lógico 1, ele entra em estado de comutação (transição) as saídas se invertem.

#### Flip-flop JK mestre-escravo

Constituído por dois Flip-flops JK, com sinais de clock invertidos, não permitindo duas variações na saída do mesmo período de clock.

#### Flip-flop tipo D

Considerado circuito lógico básico, armazena 1 bit, duas entradas e duas saídas: Clock e D (data ou delay), saídas Q e !Q, semelhante ao JK, porém o J é o D e o K também é o D, porém invertido por uma NOT.

#### Flip-flop tipo T

Formado a partir de um JK, entradas J e K curto-circuitadas, gerando uma entrada T, 0 ele mantém o último estado do Q e 1 ele inverte as saídas.

#### Contadores

Dispositivos lógicos sequenciais, realizam contagem de bits, constituídos por um grupo de Flip-flops, dependendo da quantidade e de como estão ligados são determinados módulos (número de estados) e suas sequências de conexão, podem ser caracterizados de acordo com a

sua sequência, crescente ou decrescente e de acordo com seu módulo (binário e decimal, por exemplo).

Utilizados em diversas situações, construção de temporizadores, tacômetros, relógios digitais, bobineiras, amperímetros e voltímetros.

Basicamente são classificados em duas categorias: assíncronos e síncronos.

No assíncrono, o clock apenas indica o primeiro estágio e os próximos módulos utilizam o sinal de entrada dos módulos anteriores. Os Flip-flops não mudam de estado exatamente ao mesmo tempo, não compartilham o mesmo pulso para todos os módulos, sendo assim não possuem sincronismo.

No síncrono, a entrada de clock é conectada a todos os módulos.

Contador Assíncrono - Existe um tempo de atraso inerente ao Flip-flop, não se alteram ao mesmo tempo as transições entre o sinal de clock e a saída do !Q.

Contador Síncrono - todos os módulos recebem o clock.

Registradores de deslocamento.

São circuitos compostos por arranjos/interligações de Flip-flop, finalidade armazenar e transferir informações binárias provenientes de uma fonte externa, não possuem sequência específica de estados.

Utilizados em operações aritméticas de complementação, multiplicação e divisão, em conversão de uma informação de série em paralela, bem como vários outros tipos de circuito.

Possível inserir bits de forma serial e paralela. Informação série quando os bits são apresentados sequencialmente, necessitando de apenas uma via para transferência dos bits. Informação paralela é quando os bits são apresentados simultaneamente, a transferência de informação ocorre de forma instantânea.

Possível implementar 4 tipos de registradores: Entrada e saída serial, entrada paralela e saída serial, entrada serial e saída paralela, entrada paralela e saída paralela.

O deslocamento de dados ocorre tanto para esquerda como para direita.

Se quisermos obter um registrador de deslocamento para direita é necessário alterar a ordem dos Flip-flops tipo D, a entrada do registrador será agora no primeiro Flip-flop da direita.

Controladores Lógicos Programáveis

Muito utilizados nas indústrias de manufatura, devido a sua versatilidade nas funções de controle e à sua robustez. Imunes a ruídos eletromagnéticos e resistentes a vibrações. Composto por: entrada e saída de dados, barramentos de dados, componentes de

processamento e armazenamento de informações, dispositivos lógicos e eletrônicos, circuito integrado, bem como softwares de programação (linguagem de programação).

Permitem distintas formas de programação. Principais usos na indústria: acionamento de motores, máquinas, sensores, esteiras, acionamento de sistemas hidráulicos e pneumáticos, operação de sistemas robóticos, supervisão e controle de processos industriais (vazão, temperatura e pressão), sinalização entre outras aplicações.

Principais fabricantes: Siemens, Rockwell, Toshiba, Mitsubishi, BCM, ALTUS e Schneider-Electric.

#### Linguagens de programação para CLPs

A Norma IEC 61131-3 Padronização internacional de Linguagens, Estrutura de softwares Execução de programas em controladores lógicos programáveis definem cinco tipos:

ST - Texto estruturado;

IL - Lista de instruções;

LD - Linguagem Ladder;

FBD - Diagrama de blocos;

SFC - Diagrama de fluxo.

#### Linguagem de programação ladder:

Também conhecida como lógica de diagrama de contatos, desenvolvida com mesmo diagrama de comandos elétricos. Considerada linguagem de fácil entendimento, utiliza interface gráfica.

Programar um controlador lógico por meio da linguagem Ladder é, na verdade, controlar o acionamento de máquinas e equipamentos por meio de uma combinação lógica de entradas e saídas e os contatos de entradas, utilizando contatos (chaves normalmente abertas ou fechadas) em suas entradas e relés (bobinas ou chaves) em suas saídas.

Um esquema formado por duas linhas verticais que representam analogamente os polos positivos e negativos de uma fonte de alimentação. Nelas desenham-se ramais horizontais que possuem por sua vez, chaves.

#### Barramentos de uma placa de computador.

Na área de arquitetura de computadores, conjunto de linhas que transmitem dados entre os componentes presentes nas placas dos computadores digitais.

Processador, chipset e a memória principal. Barramentos é um meio de transmissão compartilhado.

Velocidade normalmente medida em MHz, ele define a quantidade de dados que ele pode transmitir.

Barramento de dados - Permite a troca de dados enviados e recebidos, transmitindo 16, 32, 64 bits ou mais.

Barramento de endereço - Tem como função informar a origem e/ou destino dos dados provenientes do barramento de dados.

Barramento de controle - Atua como regulador de acesso e uso dos barramentos de dados e endereços, já que estes barramentos são compartilhados por todos os componentes do computador.

Podem ser considerados internos e externos.

Internos - responsáveis pela transmissão de dados entre os componentes internos da placa, externo interliga os periféricos. USB, SATA. HDMI

Barramentos utilizados:

Barramento ISA - primeira versão 8 bits, com o surgimento do Intel 80286, passou a trabalhar com 16 bits.

Barramento PCI - surgiu com o Intel no início de 1990, 32 bits. São nas cores brancas e menores que os slots ISA.

Barramento AGP - Também conhecido como porta de aceleração gráfica, foi desenvolvida pela Intel para melhorar o desempenho de placas de vídeo 3D, que utilizam o Barramento PCI. 32 bits, velocidade de 66MHz, várias cores como verde, laranja e marrom.

Barramento PCI Express - desenvolvido pela Intel em 2004 para substituir os PCI e AGP. Vários segmentos: 1x, 2x, 4x, 8x, 16x, 32x. Quanto maior o índice, maior a taxa de transferência.

### ### 3.3 Microprogramação, Microprocessador Genérico e Circuitos Integrados Periféricos

![[Untitled]](<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d0604ae7-b1ae-4119-8e85-bf3c86d3fc17/Untitled.png>)

Alem de utilizar sinais de controle, cada microoperação possui uma notação simbólica, formando uma linguagem de microprogramação.

Vantagem da microprogramação é a sua simplicidade do projeto da unidade de controle. A desvantagem é o uso da arquitetura CISC, em razão da dificuldade de sequenciar microinstruções e gerar sinais de controle.

Sequenciamento e execução de microinstruções - Unidade de controle desempenha duas tarefas básicas: Sequenciamento de microinstruções, e a execução de microinstruções.

Podem comprometer o formato da microinstrução quanto o tempo para executá-la. Existem 3 técnicas de sequenciamento de microinstruções.

Dois campos de endereço - A primeira técnica se baseia em dois campos de endereço; a segunda se diz respeito ao campo de endereço único; terceira trata do formato variável da informação.

A de dois campos é a mais fácil, existe um multiplexador para definir o destino para os campos de endereço e para o registrador de instrução. Ele pode transmitir o opcode ou um dos dois endereços para o registrador de endereço de controle. Em seguida o reg é decodificado para obter o próximo endereço de microinstrução. A seleção é feita por meio de flags. Requer mais bits na microinstrução

Endereço único - Com esta configuração, as opções para o próximo endereço são: campo de endereço, código do registrador de instrução e próximo endereço sequencial. Definido pelos sinais de seleção de endereço.

Formato variável de informação - Fornece dois formatos distintos de microinstrução, um bit define qual será o formato utilizado, o restante são usados para ativar os sinais de controle. Já no outro formato, alguns bits conduzem o módulo de lógica de desvio e os restantes fornecem o endereço.

No primeiro formato, o próximo endereço é o próximo endereço sequencial ou derivado a partir do registrador de instrução.

Segundo formato é especificado para um desvio condicional ou incondicional.

Sendo necessário um ciclo inteiro para cada microinstrução de desvio.

Execução de microinstruções - ciclo de uma microinstrução é o evento básico em um processador microprogramado.

Parte de busca determinada pela geração de endereço.

Parte de execução é geração de sinais de controle, podendo ser direcionados para o interior do processador; ou para o barramento de controle externo ou outras interfaces.

Possui também um módulo lógico de sequenciamento, contendo a lógica para gerar o endereço da próxima microinstrução. Entradas: o reg de instrução, flags da ALU, o reg de endereço de controle e o registrador buffer de controle.

Controle do módulo é realizado por um clock que determina o tempo do ciclo da microinstrução.

Microprocessador genérico (CPU - unidade central de processamento) - Capaz de acessar, controlar, e executar instrução da memória principal. Gerenciar operações de leitura e escrita e executar operações.

Componentes:

Registrador de instrução - guarda a instrução mais recente, a cada novo ciclo, faz-se a leitura na memória principal. ao final ela é armazenada no Registrador de instrução.

Decodificador de instrução - Entendida como autorização para que a CPU realize uma operação. Uma vez que existem inúmeras instruções diferentes, é necessário identificá-las. Identifica a instrução com base na instrução a ser executada.

Registros de uso geral - Utilizados para armazenar dados e endereços de memória, possuem comunicação direta com barramento interno.

Unidade de controle - Busca e interpretação de instruções, A unidade de controle gera sinais que são enviados a diversos componentes do sistema computacional, arquitetura interna e externa do microprocessador (memória principal, periféricos, chipset e barramentos de expansão)

Cache interna (L1) - Memória estática no interior do microprocessador.

FPU (Float Point Unit) - Unidade de ponto flutuante, encontrada em microprocessadores avançados, trabalha com operação em números reais.

Unidade Lógica e Aritmética ( Arithmetic Logic Unit - ALU) - Executa instruções dos programas. Constituídos por circuitos lógicos que realizam operações aritméticas (somar, subtrair, dividir e multiplicar) e lógicas (AND, NAND, OR, NOT e XOR).

Barramento Interno - Envia e transporta informações entre componentes internos do microprocessador.

Barramento de dados - troca de dados enviados e recebidos pelos componentes presentes na placa do computador, exemplo processador e memória. Podendo transmitir 16, 31, 64 128 bits. Bidirecional

Barramento de endereços - informa a origem e/ou destino dos dados proveniente do barramento de dados. Unidirecional, somente a CPU pode acionar a memória RAM para realizar operações de gravação, leitura. Exemplo barramento de endereço de 20 linhas, é possível utilizar no máximo 20 bits de endereço, sendo assim a capacidade de armazenamento na memória RAM é de no máximo 1MB, pois  $2^{20} = 1.048.576 \text{ Bytes} = 1\text{MB}$ .

Barramento de controle - controlador de acesso e uso do barramento de dados e de endereço. Bidirecional.

Arquiteturas de microprocessadores: CISC, RISC e híbrida.

CISC muito utilizadas por processadores da Intel e AMD. RISC em PowerPC (Apple, Motorola e IBM) e SPARC (SUN).

CISC - Ela já possui diversas instruções armazenadas no próprio processador. Mas como ela é implementada e armazenada em microcódigo no processador, torna-se difícil modificar a lógica de tratamento de instruções.

Capazes de executar centenas de operações complexas distintas.

Permitindo receber as instruções dos programas e executá-las por meio das próprias instruções já gravadas na sua microprogramação. Facilitando a programação de alto nível. Reduz tamanho do código e o espaço utilizado na memória pelos programas. Desempenho comprometido porque o processador deve executar instruções maiores e mais complexas. Custo elevado, grande quantidade de componentes internos no chip. Exemplos: System/360 - z/Architecture, PDP-11, o VAX, o Motorola 68k e os x86.

RISC - Custo reduzido, poucas instruções gravadas no chip, menos componentes internos, projetos mais simples, desempenho superior ao CISC.

Programador tem mais trabalho, sem microprogramação, instrução executada diretamente pelo hardware. Não tem microcódigo.

Exemplos: DEC Alpha, AMD 29K, ARC, Atmel AVR, Blackfin, MIPS, PA-RISC, Power (inclusive PowerPC), SuperH e SPARC.

Arquitetura Híbrida - Junção das duas arquiteturas, trabalha bem com instruções que requerem maior demanda, instruções de pouca complexidade e quando necessário executa instruções complexas.

Maioria dos processadores possuem arquitetura híbrida.

Circuitos integrados periféricos (CI, microchip ou chip) - constituído por milhares ou milhões de dispositivos semicondutores, como diodos semicondutores e transistores bipolares.

Construídos por diodos, transistores bipolares, amplificadores operacionais, resistores e capacitores. Algumas vantagens:

redução de custo na produção industrial;

menor peso;

dimensões reduzidas;

baixo aquecimento;

aumento da confiabilidade do sistema eletrônico;

maior velocidade de processamento do sinal (áudio e vídeo, por exemplo);

menor consumo de energia (potência), entre outras.

Classificações quanto à sua aplicação:

podendo ser classificados quanto à sua aplicação, grau de integração, tipo de encapsulamento, processado de aplicação e ao tipo de transistores e família lógica.

Circuitos integrados analógicos, produzem sinais contínuos, utilizados basicamente para amplificação de sinais. Utilizam amplificadores operacionais em suas pastilhas.

Circuitos integrados digitais - operam segundo lógica booleana. Muito utilizados em placas de computadores, sistemas microprocessados e microcontrolados, em sistemas de controle de processos industriais, em circuitos de memórias (RAM, ROM, PROM e EPROM, por exemplo).

Graus de integração - número de componentes contidos em uma pastilha de silício.

SSI (Small Scale Integrations - integração em pequena escala): 100 componentes por pastilha.

MSI (Medium Scale Integration - integração em média escala): entre 100 e 1000 componentes por pastilha. Codificadores, decodificadores, registradores e contadores.

LSI (Large Scale Integration - integração em grande escala) 1000 - 100000 componentes por pastilha. Utilizados para efetuar funções lógicas complexas, como a parte aritmética de uma calculadora eletrônica.

VLSI (Very Large Scale Integration - integração elevada em larga escala): 100000 - 10.000.000. Utilizados em sistema de computação

VULSI (Ultra Large Scale Integration - integração em escala ultra larga): maiores que 10.000.000. Utilizados em sistema de computação avançados.

Tipos de encapsulamento: serve para proteger da umidade e auxiliar na devida dissipação de calor. Pode ser metálico, cerâmico ou polimérico.

Encapsulamento com filas de pinos (DIL, QIL ou SIL): DIL (Dual Inline), a identificação dos terminais é realizada a partir do terminal 1, identificado por um pequeno orifício localizado na esquerda do chip. Inicia por aí e volta pelo outro lado percorrendo no sentido anti-horário do componente. Baixa potência.

QIL (Quad Inline) média potência, como amplificadores de áudio, permitindo maior distância entre as conexões, SIL de forma semelhante (Single Inline)

Encapsulamento plano (Flatpack): redução na espessura do invólucro. Terminais dispostos horizontalmente, menor espaço possível.



Encapsulamento metálico: cilíndrica metálica, terminais em linha, conectados em uma base, capaz de operar em altas e baixas temperaturas, terminais iniciam pela pequena marca em sentido horário com o componente visto por baixo.

Encapsulamento especial: inúmeros terminais para conectar grandes quantidades de componentes. Utilizados em placas-mãe de computadores.

#### Processos de fabricação

Preparação de substrato: preparação do substrato, pastilha de cristal de silício, formando wafers. originando centenas de circuitos impressos.

Oxidação: silício reage com oxigênio formando dióxido de silício ( $\text{SiO}_2$ );

Difusão: movimento de átomos de zona de grande concentração para baixas, dopagem do material.

Implantação iônica: método de dopagem superficial, materiais acelerados por campo elétrico se chocando na superfície do material ficam retidos na rede cristalina.

Metalização: pulverização catódica, o metal de revestimento é aquecido por meios elétricos ou químicos.

Fotolitografia: é a geração de máscaras fotográficas para as diferentes etapas de fabricação dos circuitos. Permite proteger da luz incidente zonas de um material fotoresistente depositado sobre a superfície do substrato. No processo de fotolitografia, as zonas de material fotoresistente não protegidas alteram as suas propriedades elétricas. O material fotoresistente não modificado é retirado da superfície do substrato com uma solução química adequada, obtendo-se o molde necessário para o próximo processo de fabricação. No final do processo, os circuitos integrados são encapsulados de acordo com os tipos de encapsulamento disponíveis para cada tipo de aplicação.

Encapsulamento.

#### Tipos de tecnologia

Principais parâmetros são tensões elétricas operacionais e as correntes elétricas mínimas e máximas das entradas e saídas, bem como tempo de atraso de propagação de sinais existente na comunicação entre as portas lógicas do CI.

Fan-out - Capacidade de saída de um circuito lógico, deve ser projetada para se conectar com a capacidade de entrada de outro CI. Fator de carga: quantidade de entradas que pode ser ligadas a uma única saída.

#### Tecnologia CMOS:

Utiliza transistores de efeito de campo (FET) dos tipos MOSFET e JFET. Utiliza tanto MOSFET canal tipo N quanto o tipo P, devido à complementação que um atribui ao outro na operação.

Vantagens: baixo consumo de energia, imunidade a ruído, alto nível de integração e simplicidade do projeto.

#### Família lógica TTL

Existem várias famílias lógicas: RTL, DTL, TTL, HTL, ECL e o I<sup>2</sup>L, com exceção da TTL as demais estão obsoletas.

TTL - constituída basicamente por transistores NPN e / ou PNP, diodos de junção PN e resistências difusadas. É construído de portas NAND, subfamílias TTL padrão, TTL de baixa potência, TTL de alta potência, TTL Schottky, TTL Schottky de baixa potência, TTL saída de coletor aberto, TTL saída em alta impedância.

Designação de circuitos que trabalham com 5,0 V sendo derivada da família DTL. Transistores bipolares, multiemissores. Reduz a quantidade de transistores necessários para construir as portas lógicas.

Encontrados em duas séries: uso comercial inicial com número 74. Faixa de temperatura da ordem de 0°C a 70°C.

Segunda série de uso militar, iniciando com o número 54xxx, abrangendo uma faixa ampla de -55°C a 125°C.

A tecnologia CMOS quanto a TTL são desenvolvidas com as típicas portas lógicas.

Presença de T<sub>3</sub> no coletor caracteriza o Active-Pull-UP e a sua ligação através do diodo D sobre o coletor T<sub>4</sub>, formando um elevador de potencial, Totem-pole.

Como escolher a tecnologia?

Necessário analisar características:

Consumo de energia do CMOS é menor, bom para projetos com baixo consumo, outra vantagem é o tempo de chaveamento, rapidez na troca de nível lógico, em razão dos JFETs e MOSFETs.

Porém queimam com certa facilidade, devido a eletricidade estática, necessário embalar antistática e são baratos.

TTL elevado consumo de energia em razão dos transistores bipolares, tempo de chaveamento alto, atraso de propagação.

![[Untitled]](<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fee2ee84-c0c1-449e-97ca-fafb1b832f97/Untitled.png>)

Microprocessador - Dispositivo lógico com funções de acessar, controlar e executar instruções existentes da memória principal. Para o programador a arquitetura de um processador pode ser definida pelo conjunto de instruções e o endereçamento como pelos registradores de instruções

Cada um tem seu próprio conjunto de instruções minemônicas, que representam códigos binários das instruções.

Conjunto de instruções estabelece a fronteira entre software e hardware. Permite movimentação de dados e execução de operações no Microprocessador.

Instruções de Máquina

Opcode: especifica a operação a ser realizada. representados por abreviações (minemônicos) ADD, SUB, MPY e DIV

Referência ao operando fonte: a operação pode envolver um ou mais operandos fontes, ou seja que são entradas para a operação;

Referência ao operando destino: a operação pode produzir um resultado;

Referência à próxima instrução: informa à CPU onde fazer a busca da próxima instrução.

Cálculo de endereço da instrução

Busca da instrução: realiza leitura da instrução e do seu local na memória para o microprocessador.

Decodificação da operação da instrução: determina o tipo da operação a ser realizado e os operandos;

Busca do operando: acessa o operando na memória ou realiza a leitura da E/S;

Operação dos dados: realiza a operação indicada na instrução;

Armazenamento do operando: grava o resultado na memória ou envia para E/S

Operando fonte e destino podem estar na memória principal ou virtual, no registrador do processador, no próprio campo da instrução ou no dispositivo E/S

Representação de uma instrução

16 bits, 4 Opcode, 6 Referencia ao operando, 6 Referencia ao operando.

Comun utilizar representação simbólica:

Add

SUB

MPY

DIV

LOAD

STOR

Tipos de instrução

Processamento de dados: instruções aritméticas e lógicas;

Armazenamento de dados: movimentação de dados para dentro ou fora dos regs/memo;

Movimentação de dados: instruções de E/S (entrada e saída);

Controle: instruções de teste e desvio;

Número de endereços

Quantidade de endereços por instrução geralmente 1, 2 ou 3, quanto mais endereços, instruções maiores mais complexas, precisando de mais regs, quanto menores mais simples e primitivas, menos regs.

Lei de Moore

Primeiro microprocessador comercial foi o Intel 4004, fabricado em 1971, 4bits, 2300 transistores, memória 640 bytes. Surge depois 22 anos Intel pentium, 3,1 milhões de transistores, clock de 60MHz, versões 75, 90, 120, 133, 150, 166, 200 e 233 60MHz

2000 surge pentium 4, 42 milhões de transistores, clock variando entre 1300 e 3800 MHz em 2006, depois mudando o nome para Core, Core Duo com 152 milhões de transistores, 1600 e 3200 MHz, evolui para o i3, i5, i7 e i9.

Afirmção básica da lei de Moore. O poder de processamento dobra a cada dezoito meses em razão do processo de integração de transistores.

#### Técnicas de programação Assembly

Considerada linguagem de baixo nível. Tradução do assembly para binário é realizada pelo assembler.

#### Vantagens:

Otimização do código para aproveitar o máximo das características intrínsecas do hardware

Existência de assemblers para todos os microprocessadores.

Otimização de espaço na memória;

Rapidez na execução do código.

#### Desvantagens:

Maior tempo na decomposição de cada tarefa em conjunto de instruções e/ou operações

Maior conhecimento de programação por parte do programador;

Condição desproporcional entre o conjunto de instruções e as tarefas que o microprocessador normalmente executa;

Não apresenta tanta portabilidade, ou seja, a linguagem é portável para uma família de microprocessadores e não para um sistema de trabalho;

Maior custo de desenvolvimento, tendo em vista a baixa expressividade entre os profissionais da informática.

#### Programando em Assembly

##### Ambiente de programação MPLAB

Labels: designação textual de uma linha em um programa ou o início de um conjunto de linhas, Max 32 caracteres, inicia com letra do alfabeto ou \_.

Instruções;

Operandos - Registros, variáveis e constantes, constantes são designados por "literais";

Diretivas - característica inerente à própria linguagem de programação. exemplo, NIVEL pode ser uma designação para uma variável localizada no endereço 0Dh na memória principal.

Comentários;

## Introdução a microcontroladores

### Diferença entre microprocessador e microcontrolador?

Diferen-se quanto a funcionalidade - para que o microprocessador execute funções, necessita de outros componentes externos.

Por outro lado o microcontrolador, pensado para integrar inúmeros componentes em um único CI.

Definição: circuito integrado programável utilizado no controle de processos que exigem uma lógica computacional, como automação de indústrias, encontrados em elevadores, impressoras.

### Microcontroladores PIC

Circuito integrado que possui inúmeros componentes internos. Integram a família TTL ou mesmo como circuito integrado CMOS. Dispõem de CPU, PROM, RAM, Linhas de E/S, barramentos, clock, registrador, contador entre outros.

Possuem algumas famílias:

PIC 8bits - PIC 10-155 microcontroladores (8 a 64 pinos) e PIC18-223 microcontroladores (18 a 100 pinos). Exigem economia de energia, alta compatibilidade e robustez.

PIC 16bits - PIC24, PIC3, PIC33. integram 432 microcontroladores (12 a 122 pinos) conversores AD/DA maior resolução com funções de processamento digital de sinal, bem como vários recursos, como USB e protocolos TCP/IP.

PIC 32bits - PIC32 com 184 microcontroladores (28 a 144 pinos) Família dedicada a operações com pontos flutuantes. até 200MHz (48MHz das outras famílias) tem interface gráfica, áudio e controles para realizar multitarefas.

Existem vários compiladores, necessitam de ambientes de programação distintos, desenvolvidos pela Microchip Technology Inc.: MPLAB XC8, MPLAB XC16 e MPLAB XC32. Suportam SO, Linux, MacOS e Windows, compatíveis com versões anteriores do MPLAB IDE.

Registrador W diretamente ligado a ULA.

Memo de dados conectada a um barramento de 8bits, memória flash conectada a um barramento de 14 bits. Também existem alguns blocos representativos, os quais são circuitos de memória EEPROM, timers comparadores e portas série. Nas portas A e B, existe um diagrama de pinos do microcontrolador. Encapsulamento tipo DIP, SSOP, e FN. Frequência

maxima do clock 20MHz. ou por um oscilador interno de 4Hz ou 37kHz. clock dividido internamente por 4.

Arduino - plataforma de prototipagem. Projetada com microcontrolador Atmel Corporation AVR (microcontrolador RISC 8bits), normalmente ligauem C++ ou C.

Utilizam serie de chips megaAVR, ATmega8, ATmega168, ATmega328 e ATmega1280.

## ## 4 Noções de Inteligencia artificial.

### ### 4.1 \*\*\*\*Introdução e Aplicações da Inteligência Artificial\*\*\*\*

Inteligencia artificial recebeu influencia de diversas áreas do conhecimetno desde a sua origem.

Incluindo Matematica, economia, neurociencia, psicologia e filosofia, egenharia de computadores e teoria de controle.

Matematica - logica, representação formal e as provas a computação agoritimos e probabilidade.

Economia - Proposição de diversas teorias e decisões racionais, tomadas de decisões em situações de incerteza, teoria dos jogos e processos de decisão de Markov.

Neurociencia - Estudo do funcionamento do cerebro e estudos comparativos entre o cérebro e os computadores.

Psicologia - questionamentos sobre como agimos e pensamos.

Filosofia - Aristóteles, os filosofos foram os grandes alicerces da Inteligencia Artificial. Diversos movimentos filosoficos contribuíram para o desenvolvimento como o Racionalismo, DUalismo, Materialismo, Empirismo.

Arquitetura de Computadores - como constuir computadores podesos que podem funiconar com as estrategias propostas para a INteligencia Artificial.

Teoria de controle e cibernética - Desenho de agentes inteligentes que recebem percepções do ambiente, garantinado e maximizando o desempenho de sistemas que buscam uma solução em escala temporal.

Lingustica - recalionamento entre a linguagem e o pensamento, originando a função entre a linguistica mderna e a inteligencia Artificial.

## Principios da inteligencia artificial.

def: Esudo e projeto de agentes inteligentes, agent eé um sistema que percebe seu ambiente e age de tal forma a maximizar sua probabilidade de sucesso.

Por que estudar inteligencia artificial?

Nos libertara das amarras mentais.

Quatro escolas do pensamento para justificar e Inteligencia artificial.

Pensando Humanamente - abordagem cognitiva que necessita saber como o ser humano pensa.

Agindo Humanamente - Teste de Turing.

Pensando racionalmente - As escolas gregas desenvolveram notações e regras derivadas do pensamento. Vem com alguns problemas consequentes, nem tudo pode ser resolvido puramente com logica.

Agindo racionalmente - trata-se de fazer a coisa certa, agir para maximizar a chance de realizar um objetivo.

Principios gerais de agentes racionais.

Componentes para construi-los.

Lista de contribuições:

Aristóteles,

Babbage,

Boole,

Alan Turing,

Jhon Mc Varchy,

Warren McCulloch,

Walter Pitts,

Claude Shanno,

Marvin Minsky,

Dean Edmonds,

Herbert Simon,

Allen Newell,

David Waltz,

Tom Mitchell,

Stuart J. Russel,

Peter Norvig



Gestão por competência.

Fatos relevantes:

1940 - 1950 Mulloch e Pits - circuitos booleanos para modelar o cérebro; Máquina computacional de Turing, Teste de Turing e jogo da imitação.

1950 - 1970 Uso da inteligência artificial em jogos de xadrez e damas; o nascimento da inteligência Artificial da reunião do Dartmouth College em 1956

1970 - 1990 Período dos sistemas especialistas; aplicações industriais.

1990 em diante - retorno das redes neurais;

conceitos mais precisos de agentes inteligentes.

Inteligência Artificial torna-se a ciência e a probabilidade passam a ser fortemente empregada para modelar as incertezas do mundo real.

Usos na indústria:

reconhecimento de voz;

reconhecimento de caligrafias;

tradução de máquina;

robótica;

sistemas de recomendação

filtros de correio eletrônico;

detecção de características do rosto humano;

detecção de células cancerígenas

algoritmos poderosos em máquinas de jogos;

carros autônomos;

Agentes inteligentes - de modo geral é algo que age/reage mediante estímulos que recebe.

Percebe por meio de sensores, e por meio de atuadores, tomar uma ação. Agente é o elemento dotado de inteligência.

Ambiente - espaço no qual o agente atua em análises posteriores.

Sensores - sentem os estímulos;

Agente;

Atuadores - capacidades que o agente tem de provocar mudanças no ambiente ao seu redor, como braços, músculos, rodas, válvulas, motores.

Detalhando o agente - o agente age com base em seu algoritmo e percorre o que chamamos de espaço de estados, até que encontre a solução para um dado problema.

O agente não toma decisão por ele próprio sobre a correção das ações tomadas, responsabilidade transferida para o ambiente, para o programador, não há conhecimento prévio do ambiente e as ações que ele for capaz de realizar não são racionais.

Com base em um aspirador de pó inteligente, percebe-se se há sujeira no ambiente. liberdade para se mover mediante seu algoritmo, pode limpar a sujeira ou não fazer nada.

Necessita de indicador de desempenho.

Aspectos a serem observados para classificar o aspirador de pó como agente racional são: a complexidade do ambiente, adequação dos sensores ao ambiente, atuadores presentes, indicador de desempenho, um algoritmo coerente com todos os aspectos anteriores.

Um agente do mundo real, deveria estruturar a sequência completa de ações a tomar, o que quase nunca ocorre na prática, quase nunca sabemos, a priori, as percepções que nosso agente encontrará no mundo real.

Se não podemos conceber para o nosso agente todas as respostas simplesmente porque não conhecemos todas as perguntas, as técnicas tradicionais de programação embarcadas, por exemplo, aquelas empregadas na Engenharia de Software, não servirão aqui.

New Map, Select Map: estas opções permitem, respectivamente, criar um ambiente ou selecionar um ambiente já existente. DoOneStep, DoOneRun, NextRun, DoAllRun: execução que significa, respectivamente, um passo, uma rodada, próxima rodada e todas as rodadas simultaneamente. Steps, Time: definições da quantidade de passos e do tempo de duração desses passos. DirtyDegree: grau de sujeira, mais uma vez, interferindo e exemplificando o ambiente mencionado em nossa abordagem teórica.

Medida de desempenho, energia consumida pelo agente.

Código parcial em c++

Quanto mais complexo o ambiente, mais complexo o código do nosso agente.

Todo agente pode melhorar o desempenho à medida que puder aprender com suas experiências.

Outros exemplos de agentes:

Em carros autônomos:

- Ambiente: estrada, demais carros, pedestres e sinalização.
- Atuadores: acelerador, freio, setas e buzina.
- Sensores: odômetro, velocímetro, acelerômetro e GPS.
- Indicador de desempenho: segurança da viagem, tempo gasto no percurso, respeito às leis de trânsito e conforto.

Para o agente aspirador:

- Ambiente: salas, mesas, piso de madeira, obstáculos diversos.
- Atuadores: rodas, escovas, os extratores a vácuo.
- Sensores: câmeras, detectores de sujeira, infravermelhos (detecção de obstáculos), de vácuo.
- Indicador de desempenho: eficiência da limpeza, distância percorrida, nível de uso da bateria.

Tipos de ambiente:

Completamente observável x parcialmente observável - no primeiro caso, os sensores dos agentes permitem que eles tenham acesso ao estado completo em cada instante de atuação. Caso contrário, sem estes sensores, o agente navegará em um ambiente parcialmente observável.

Determinístico x estocástico - no primeiro caso, o próximo estado do ambiente fica completamente determinado pelo estado corrente e pela ação executada pelo agente. Caso contrário, navegará em um ambiente estocástico, ou seja, probabilístico.

Episódico x sequencial - no primeiro caso, a ação para cada estado ou episódio depende somente do próprio episódio. Caso contrário, se o agente conseguir guardar as ações anteriores, será sequencial.

Agente único x multiagente - no primeiro caso, um agente resolve um problema sozinho, sem ser influenciado pelo comportamento de um segundo, eventual, agente. Caso contrário, de forma geral, se ação do oponente interfere, considera-se agente multiagente.

Conhecido x desconhecido - no primeiro caso, conhece-se completamente todas as ações possíveis em relação ao ambiente. Caso contrário, para um ambiente desconhecido, o agente precisará conhecê-lo primeiramente antes de tomar as ações.

Contínuo x discreto - refere-se ao estado do ambiente, às percepções e ações do agente, por exemplo, o cenário do ambiente altera-se com o tempo? Se sim, então, estaremos em um ambiente contínuo.

Estático x dinâmico - no primeiro caso, uma vez visualizado o ambiente, ele permanecerá o mesmo, enquanto o agente "pensa", são mais fáceis de serem manipulados.

Exemplos clássicos de agente instigentes que podemos imaginar a complexidade do algoritmo em vista complexidade do seu ambiente.

- a. Reconhecimento de voz
- b. Jogos
- c. Antivírus
- d. Planejamento estratégico em situações de guerra
- e. Assistentes pessoais virtuais
- f. Bancos de Dados
- g. Agricultura
- h. Medicina
- i. Robótica

Técnicas de resolução de problemas, começando pelas mais simples:

Regra de produção ou regra se-então - exemplo de situação:

O problema proposto envolve dois vasilhames de água, um com capacidade de quatro litros e outro com capacidade de três litros. O objetivo é descobrir como colocar exatamente dois litros de água no vasilhame de quatro litros. Uma abordagem sugerida é criar um conjunto de pares ordenados  $(x, y)$ , onde  $x$  representa a quantidade de água no vasilhame de quatro litros e  $y$  representa a quantidade de água no vasilhame de três litros. O estado inicial é  $(0, 0)$  e o estado meta é  $(2, n)$ , onde  $n$  pode variar de 0 a 3. O texto menciona suposições necessárias, como a capacidade de encher os vasilhames, despejar água no chão e transferir água entre os vasilhames. Além disso, destaca a dependência desse tipo de problema em considerações como configurações dos objetos, espaço de estados, estado inicial, estado objetivo, conjunto de regras e estratégia de controle para alcançar o estado meta.

Pode ser estudado com abordagem em árvore de busca: estado inicial - nó de partida, estados intermediários, por fim o nó meta

Outras abordagens: busca em profundidade, busca em amplitude, o menos-custo em primeiro lugar, heurística, A\*, redes neurais, lógica de fuzzy, redes Bayesianas, dentre outras.

Se se preocuparmos com repetição de nós, podemos gerar um grafo direcionado a partir da árvore de busca, gerando ciclos com a vantagem de refuzir o esforço que é gerado explorando o mesmo nó diversas vezes

Reconhecimento de padrões sensoriais é caracterizado pelo reconhecimento de padrões como uma classe de problemas especialmente aceita à inteligência Artificial.

Sistemas de preenchimento automático, presente nos celulares e computadores pessoais.

Campos que se beneficiam:

Agricultura, com o uso das IoT, análise preditiva, melhorando o planejamento e desempenho na produção agrícola;

Transporte e mobilidade, carros autônomos e auxiliados por GPS.

A área de saúde tem adotado tecnologias baseadas em Inteligência Artificial para auxiliar nos diagnósticos. Um exemplo mencionado é o caso do computador da IBM chamado "Watson", que está sendo utilizado pelo centro de pesquisas em câncer Memorial Sloan-Kettering. O Watson tem ajudado no diagnóstico de câncer e, de acordo com o autor, 90% dos enfermeiros que utilizam essa tecnologia seguem suas sugestões.

Cadeia de fornecimento, auxiliando na previsão de necessidade de insumos e da automatização da logística.

No contexto dos agentes inteligentes, existe uma diferença entre o mundo real e o mundo virtual. A modelagem do problema é necessária para simplificar a situação de forma que o agente possa manipular as possibilidades dentro dos recursos computacionais disponíveis. No entanto, essa simplificação limita as opções do agente ao universo modelado. Enquanto em jogos como as damas essa modelagem não apresenta problemas, em casos do mundo real, como o reconhecimento de imagens de gatos por uma Inteligência Artificial, pode haver desafios. A modelagem utiliza um número limitado de imagens para treinar o agente a reconhecer outras imagens de gatos, mas o mundo real oferece uma variedade infinita de possibilidades. Isso pode levar o agente a cometer erros de identificação, pois lhe falta a experiência rica e detalhada que os seres humanos possuem. Uma técnica que busca superar esse obstáculo é o aprendizado por reforço, que funciona bem quando o agente inteligente interage fisicamente com o ambiente.

A atuação dos agentes inteligentes é efetiva em problemas específicos, com escopo limitado e um número finito de estados. No entanto, essa efetividade diminui à medida que o problema se torna mais amplo, exigindo mais recursos para manter um desempenho efetivo em algum grau. Com as técnicas disponíveis atualmente, chega-se a um ponto em que os benefícios

adicionais em termos de desempenho se tornam cada vez menores, tornando inviável a generalização do agente para problemas maiores.

### ### 4.2

### ### 4.3 Aprendizagem

#### Aprendizagem de máquina

Importantes definições:

a. Tarefa: Refere-se ao problema que precisa ser solucionado ou para o qual o aprendizado é desejado. Os dados e objetivos da tarefa são fundamentais para o processo de aprendizado.

b. Representação: É a metodologia utilizada para descrever o problema e as operações envolvidas. Exemplos incluem o uso da lógica de predicados na forma de cálculo sentencial.

c. Espaço de conceitos: É o conjunto de dados obtidos por meio de operações de generalização e especialização relacionadas à tarefa a ser realizada.

d. Heurística: Geralmente faz parte da solução desses problemas e está associada a abordagens não determinísticas, oferecendo orientações ou estratégias para resolver a tarefa.

e. Aprendizado: É o resultado esperado ao resolver a tarefa proposta. Envolve classificar, filtrar e identificar o conhecimento adquirido por meio de técnicas de aprendizado de máquina, representado por diferentes conceitos (C1, C2, C3...).

Aprendizado envolve generalização a partir da experiência, também pressupõe a realimentação das ações (feedback), para avaliar a qualidade dos resultados alcançados.

Existem três tipos clássicos de realimentação em seu processo de aprendizagem:

a. Aprendizagem supervisionada: ocorre a partir de exemplos, e as respostas corretas para cada um são conhecidas. Neste caso há a presença de um "professor" orientando o agente;

b. Aprendizagem não supervisionada: ocorre sem exemplos, e as respostas corretas não são conhecidas. Não há um "professor" orientando o agente inteligente;

c. Aprendizagem por reforço: o agente precisa explorar o ambiente ao redor em busca das ações com maior recompensa.

Outra forma de explicar o aprendizado de máquina é dizer que se trata de um modelo matemático incorporado a um sistema qualquer, que toma decisões com base em evidências na presença de incertezas, com a intenção de aprender com elas.

Exemplos: filtros de spam, previsões de ações em bolsa de valor, sistema de recomendação de marketing, sistema de reconhecimento de voz e da face e etc.

Conhecendo os principais representantes

Sistemas especialistas são projetados para dotar agentes inteligentes com regras de produção baseadas na experiência de especialistas humanos, como médicos, engenheiros, mecânicos de automóveis, instrutores de voo, entre outros. Esses sistemas refletem o conceito de um professor orientando as decisões do agente, fornecendo conhecimento especializado para guiar suas ações.

Redes neurais artificiais são modelos computacionais compostos por elementos simples que atuam de forma interconectada, transmitindo ou não sinais digitais (0 ou 1) com base em funções que combinam o sinal de entrada com limiares pré-estabelecidos.

Algoritmos genéticos são inspirados pelos princípios da evolução biológica e aplicados a problemas complexos. Esses algoritmos geram conjuntos de soluções candidatas e incentivam a sobrevivência dos conjuntos mais promissores, eliminando os menos eficientes. Novas soluções candidatas são geradas a partir dos conjuntos sobreviventes, em um processo contínuo de adaptação. Essa abordagem tem semelhanças com a teoria da evolução das espécies de Charles Darwin.

Sistemas Especialistas

Interface: meio de comunicação do usuário com a máquina, perguntas e respostas, menus.

Mecanismos de inferência: Escolhe as regras mais adequadas ao problema, com base em um conjunto de regras.

Base de conhecimento: Divide-se entre base especialista ou específica do problema. Suas regras geralmente na forma de "se (condição), então (conclusão)", são essenciais para a qualidade do desempenho da máquina como um todo.

Sistemas especialistas são de custo elevado e, utilizam diretizes que permitam avaliar a real necessidade de sua aplicação.

## Redes neurais

Tambem conhecidas como: Redes conexionistas, processamento paralelo distribuido e computação neural, modelos matematicos desenvolvidos com base em neuronios.

Neuronios anatomia:

- a. O corpo celular é a região onde o núcleo do neurônio está localizado.
- b. Os dendritos são extensões ramificadas do corpo celular que se conectam aos terminais sinápticos de outros neurônios, formando as sinapses.
- c. O axônio é uma extensão que parte do corpo celular e conduz os sinais elétricos até os terminais sinápticos.
- d. Os terminais sinápticos são as terminações do axônio que se conectam aos dendritos de outros neurônios, estabelecendo as sinapses.

Capaz de disparar corrente eletrica atravez dos axônios quando atigem determinado limiar de excitação energetica, e se conecta a ourtos neuronios atraves de seu dendritos.

Partem da premissa o niver deste fluxo seja o indicador de nivel de aprendizado da rede.

Aplicações comuns: classificação, reconhecimento de padrões, predição, otimização, filtraem de ruídos e os assistentes visuais.

## Neurônio Booleano

Trata-se de um modelo computacional capaz de receber diversas entradas e atribuir-lhes pesos, processar-las, comparar o resultado obtido com um valor pré-determinado, seu limiar, e gerar um valor de booleano de saída, podendo apresentar uma de duas respostas possíveis

Se o resultado da comparação for maior que o limiar, a saida é 1

Se não, sera 0;

O resultado é obtido pela soma dos produtos de entrada por seus respecticos pesos e, matematicamente representando, teremos:

Resultado= Somatorio de 1 ate N (Entrada i \* pesos i)



Sáida é uma unção do resultado.

$saida = f(resultado) \begin{cases} 0 & \text{se } resultado \leq Limiar \\ 1 & \text{se } resultado > Limiar \end{cases}$

Linha 1: Resultado =  $0 \times 1,1 + 0 \times 1,1 = 0$ ; resultado menor ou igual ao limiar, portanto, valor de saída = 0;

Linha 2: Resultado =  $0 \times 1,1 + 1 \times 1,1 = 1,1$ ; resultado maior que o limiar, portanto, valor de saída = 1.

Linha 3: Resultado =  $1 \times 1,1 + 0 \times 1,1 = 1,1$ ; resultado maior que o limiar, portanto, valor de saída = 1;

Linha 4: Resultado =  $1 \times 1,1 + 1 \times 1,1 = 2,2$ ; resultado maior que o limiar, portanto, valor de saída = 1.

Nos de entrada (E1 a E4): podem converter sinais analogicos em digitais. Considere a conversao de urn sinal analogic° de temperatura de um sensor termopar em urn sinal digital proporciona;

Nos escondidos (CE): responsaveis pelo processamento interno da rede RNA e, embora nao sejam essenciais, sua quantidade e numero de camadas con-ferem major precisao aos resultados obtidos pela rede após o treinamento;

Nos de saida (S): trata-se da Ultima camada de processamento da rede, que apresenta os valores binarios já mencionados anteriormente. Este processo tamb4m é conhecido como o recall da rede; Conexoes: sejam de entrada ou de saida, sao as interligacties entre os nos da rede;

Pesos (P1, P2, ...): fatores de multiplicacao para cada um dos sinais de entra-da dos nos;

Funcao de processamento ou ativacao ( $f(x)$ ): funcao que combina o par (sinal de entrada, peso) dentro do no;

Limiar (L): valor arbitrario gerado para cada no, posteriormente comparado corn o resultado da somat6ria ponderada, conforme explicado anteriormente.

Aprendizado: ajustes de pesos e limiares (funcao de ativacao) corn o objeti-vo de reforcar os estados corn respostas corretas e inibir os demais estados. Como já sabemos, trata-se de algoritmos heurísticos, corn inUmeros passos de tentativa e erro. Futuramente, a rede treinada passara a dificultar o reapareci-mento das respostas erradas, privilegiando o comportamento mais adequado.

Utilização: onde se dá o trabalho efetivo da rede na solução do problema para o qual foi apresentada.

Perceptron - elemento de software e/ou hardware capaz de aprendizado supervisionado, cuja função é um classificador binário.

Alimenta-se sobre a função linear de previsão, como uma regressão linear.

Classificação -, formas com características diferentes em processo de separação por algoritmo Perceptron.

Previsão - pontos em um gráfico geram uma reta por regressão linear, aproximando os resultados de um experimento por uma função linear

Perceptron recebe  $n$  entradas  $X$ , com peso  $W$  em cada uma. Expressão o conjunto de entradas:  $[x_1, x_2, x_3, \dots, x_n]$ . Também um limiar  $L$ , perceptron gera na saída uma  $f(x)$ , dada pela fórmula:

$$f(x) = \begin{cases} 0 & \text{se } W \cdot X + L \leq 0 \\ 1 & \text{se } W \cdot X + L > 0 \end{cases}$$

$$\{1 \leftarrow W \cdot X + L > 0\}$$

$W \cdot X$  produto vetorial dos pesos  $[w_1, w_2, w_3, \dots, w_n]$  e entradas  $[x_1, x_2, x_3, \dots, x_n]$ :

$$W \cdot X = \text{Somatório de } W \cdot X = \sum_{i=1}^n (W_i \cdot X_i) \text{ com } i \text{ de } 1 \text{ até } n$$

Os perceptrons quando conectados a uma rede neural com uma camada de entrada e de saída, podem ser configurados com uma regra simples de aprendizado, regra de Rosenblatt, gera por meio de produtos vetoriais  $W \cdot X$  a função linear que separa os elementos de um conjunto.

Regra diz que os pesos devem ser corrigidos a cada iteração do sistema com um par [Entrada, rótulo].

Funcionando apenas para conjuntos separáveis.

## Multicamadas

Usada em situações que é necessário classificar um determinado conjunto, no qual não é possível aplicar a classificação linear, no Perceptron não é possível representar um XOR.

RNA podem apresentar várias estruturas: Hopfield, Adaline, Madaline, CCN, Máquinas de Boltzmann, Kohonen e outras.

No Perceptron multicamadas, tem-se um conjunto de Perceptrons arranjados em diversas camadas. consegue resolver problema de classificação não linear, porem cada camada ainda é um rede linear.

O algoritmo de retropropagação, também conhecido como backpropagation ou Regra Delta Generalizada, é utilizado em redes MLP (Multi-Layer Perceptron) com alimentação direta (feedforward). Esse algoritmo trabalha com uma função de saída que minimiza a soma dos quadrados dos erros, considerando a média sobre o conjunto de treinamento.

As redes MLP possuem duas fases: a fase de alimentação direta (feedforward), em que os dados se propagam da camada de entrada para a camada de saída, e a fase de retropropagação, na qual os valores de saída da rede são comparados com as respostas esperadas. Os erros produzidos são propagados de forma inversa (retropropagação) da camada de saída até as camadas de entrada, permitindo a atualização dos pesos e o ajuste da rede.

O algoritmo de retropropagação utiliza funções de ativação não-lineares e estabelece um valor de erro predefinido em sua saída, o que o torna mais eficiente em comparação com o algoritmo Perceptron MLP em termos de aprendizado. Em resumo, o algoritmo de retropropagação é uma melhoria em relação ao seu antecessor, oferecendo uma vantagem significativa no processo de aprendizado.

As pesquisas de Donald Hebb sobre aprendizado resultaram na expressão "aprendizado hebbiano" ou Lei de Hebb. Essas pesquisas mostraram que as conexões sinápticas mais utilizadas facilitam a transmissão dos sinais elétricos, enquanto as menos utilizadas não facilitam. Esse mecanismo, aplicado a grandes grupos de neurônios, permite o processo de aprendizado, promovendo a repetição das ações vantajosas e inibindo as menos vantajosas.

Ao discutir os processos de aprendizado em redes neurais, é importante mencionar que várias técnicas foram desenvolvidas ao longo da história. Além do aprendizado hebbiano, existem outras abordagens, como o aprendizado por correção de erro, que é usado na retropropagação do Perceptron; a Regra Delta Generalizada; aprendizado por memória; aprendizado competitivo; aprendizado de Boltzmann; e aprendizado probabilístico, entre outros. Essas técnicas contribuem para o potencial das redes neurais e têm sido exploradas ao longo do tempo nesta disciplina fascinante.

#### Algoritmo genetico

Existem diferentes abordagens para eliminar candidatos fracos em um algoritmo, como o uso de probabilidade ou uma porcentagem fixa para eliminação. O texto destaca a linha do algoritmo que menciona a substituição com base na aptidão, também conhecida como Monte Carlo, Seleção Proporcional à Aptidão ou Seleção por Roleta em literatura especializada.

O crossover, também chamado de recombinação, envolve tomar dois candidatos e trocar seus componentes para gerar dois novos candidatos. Essa ideia é destacada na Figura 13.

A mutação envolve fazer alterações aleatórias em um único candidato, como alterar um bit no padrão de forma aleatória. A mutação é importante para evitar a exclusão de componentes essenciais na solução inicial.

#### ### 4.4 Lógica Nebulosa

**Fuzzificador:** O fuzzificador transforma uma entrada numérica em um valor fuzzy. Isso pode ser feito por meio de uma função ou de uma tabela. No gráfico da função de pertinência, é possível descobrir o valor de  $\mu$  (grau de pertinência) para o valor de entrada correspondente.

**Inferência fuzzy:** Na inferência fuzzy, para cada regra, é estabelecida uma região no gráfico de saída resultante com base na pertinência de cada entrada. Geralmente, os resultados parciais fornecidos por cada regra são agregados em um único gráfico, muitas vezes por meio da união dos resultados. Se o sistema tiver entradas e saídas fuzzy, é mais simples, pois uma tabela com as regras pode ser utilizada para realizar a inferência.

**Desfuzzificador:** O desfuzzificador calcula uma saída numérica a partir do gráfico resultante da inferência. Existem vários métodos disponíveis, sendo o mais comum o do baricentro (C.G), que determina o centro de gravidade do gráfico para obter o valor de saída numérica.

Exemplo: controle de nível de água:

Dada um sistema especialista fuzz com uma única regra que controla o nível da água em um tanque por meio da abertura de sua válvula de escape, uma regra: se o "nível da água estiver alto", então abrir a válvula e sua função de pertinência

**Problema:** O objetivo é determinar a gorjeta a ser dada ao garçom com base na qualidade da comida e do serviço prestado pelo estabelecimento, seguindo as seguintes diretrizes:

- a. Se a qualidade da comida for ruim ou o atendimento for ruim, deixar uma gorjeta pequena, próxima de 5%.
- b. Se o serviço for mediano, deixar uma gorjeta mediana, próxima de 10%.

c. Se a qualidade da comida ou o atendimento forem surpreendentes, deixar uma gorjeta generosa, próxima de 15%.

A escala de graduação para a comida e o serviço varia de 0 a 10.

A solução será implementada no MATLAB, seguindo os passos a seguir.

Indução de regras:

Nesta etapa, utilizamos a lógica fuzzy para definir as regras da base de conhecimento do sistema, o que determina o comportamento do sistema como um todo. Essa etapa é chamada de indução de regras e está associada ao bloco "máquina de inferência" no modelo apresentado na Figura 3. O MATLAB oferece um editor de regras específico para essa finalidade, que é uma ferramenta de leitura apenas (read only) e possui recursos adicionais.

Visualizar graficamente, como ficarão as regras e a superfície de decisão para um dado problema;

Diagnosticar as tendências resultantes das aplicações das regras fuzzy para a solução do problema.

Árvore de decisão

A árvore de decisão é uma abordagem comum e amplamente utilizada no processo de aprendizagem. Segundo Russel e Norvig (2013), ela é considerada uma das formas mais bem-sucedidas de algoritmos de aprendizagem e pode apresentar semelhanças com as abordagens fuzzy discutidas até o momento.

Em uma árvore de decisão, um objeto ou situação é descrito por um conjunto de atributos e a árvore retorna uma decisão ou valor de saída previsto com base nessas informações.

Para evitar a explosão exponencial do número de árvores de decisão possíveis, é necessário restringir o conjunto de árvores geradas. Isso é especialmente relevante quando lidamos com um grande número de atributos.

Uma abordagem para restringir o número de árvores é selecionar o atributo mais significativo como a raiz das subárvores geradas. Por exemplo, no problema do restaurante mencionado anteriormente, o atributo "público" é mais significativo do que o atributo "tipo", pois divide os subconjuntos resultantes de forma mais eficiente, como ilustrado na Figura 19.

Essa restrição é importante para lidar com as limitações de recursos computacionais e tornar o algoritmo de árvore de decisão mais viável, mantendo sua simplicidade.

### O Algoritmo de decisão

Com base nessa abordagem, podemos propor o seguinte algoritmo para escolher o melhor atributo e, conseqüentemente, reduzir a quantidade de árvores:

Se houver exemplos positivos e negativos, escolha o melhor atributo para dividi-los.

Se todos os exemplos restantes forem positivos ou todos forem negativos, podemos terminar e responder com um valor sim ou não.

Se não houver mais exemplos, isso significa que nenhum exemplo desse tipo foi observado. Nesse caso, retornamos um valor padrão calculado com base na classificação majoritária do nó pai.

Se não houver mais atributos, mas ainda houver exemplos positivos e negativos, isso indica um problema. Isso significa que esses exemplos têm a mesma descrição, mas classificações diferentes. Isso pode ocorrer quando há ruído nos dados, ou seja, algumas informações estão incorretas. Nesse caso, podemos usar uma votação majoritária como saída.

Esse algoritmo ajuda a tomar decisões sobre a divisão de atributos e trata casos em que os dados podem estar incompletos ou com ruído, fornecendo uma solução baseada na classificação majoritária.

![[Untitled]](<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b2223ea0-7fd4-42de-b422-8e367959b884/Untitled.png>)

### ### 4.5 \*\*\*\*Introdução aos Sistemas Inteligentes e Aprendizado de Máquina\*\*\*\*

Vizualiza-se uma maior facilidade na extração de informações contidas nas obtenções de dados quando há uma pequena quantidade de variáveis.

Em suma a redução de dimensionalidade se fundamenta na busca por um conjunto de menor variáveis calculadas a partir dos dados originais.

Podem ser divididos em dois tipos de processos: processo de extração de atributos, criam novas características; processo de seleção de atributos, os algoritmos, selecionam, segundo determinado critério, o melhor subconjunto do conjunto de características originais.

Podem ocorrer de novas características não terem significado isico.

Em uso excessivo o algoritmo pode perder a capacidade de discriminar.

Tres tecnicas de extração de atributos:

Análise de componentes principais (PCA)

Escala multidimensional (MDS)

FastMap.

Os algoritmos de seleção estão embutidos nos aprendizados de máquina.

Aplicabilidade:

Dentre as áreas pode-se destacar: estatística multivariada, a teoria da informação e o reconhecimento de padrões.

Utilizada em:

Sistemas de recomendação;

Modelagem de tópicos e pesquisa de documentos similares;

Análise de imagens;

Gerenciamento de riscos;

Rede Neurais (Perceptron, Adaline Backpropagation)

RNA - Composto de diversas unidades simples que trabalham paralelamente e distributivamente. Seu processo de aprendizagem é por reforço.

Perceptron - Mais simples que a RNA, composta apenas de uma camada neural com um neurônio artificial. Principal propósito mapear o comportamento das entradas e saídas.

Backpropagation - Algoritmo de aprendizagem usa a regra delta, utilizada para treinar redes multicamadas.

Regra delta - diferença entre o erro quadrático entre os valores de entrada e saída em relação a uma amostra  $x$  de treinamentos.

Perceptron Múltiplas Camadas - tem ao menos uma camada intermediária escondida. Tendo no mínimo duas camadas de neurônios. Algoritmo utilizado no aprendizado: Retropropagação de erros ou backpropagation/ regra delta generalizada.

Usado no reconhecimento de padrões, aproximação universal de funções, previsão de séries temporais, otimização de sistemas de identificação e controle de processos.

Arquitetura

Multicamadas precisam ter:

$N$  camadas intermediárias;

M neurônios: o número de neurônios é muito importante, quantidade de neurônios em cada camada define a complexidade da rede; esta é diretamente proporcional à capacidade da rede em resolver problemas.

#### Redes Kohonen

Mapa auto-organizável de características, contém duas camadas de neurônios. Primeira detém função de receber os padrões de entrada, já a segunda a função de gerar os padrões de saída.

A camada de neurônios de entrada são completamente interconectados aos neurônios da segunda camada (competitiva). A camada competitiva apresenta arranjo organizacional que depende do objeto determinado para o mapeamento e em caso de a mesma apresentar uma rede de duas dimensões não há nenhuma imposição a forma topológica.

Pode ser definido como: rede neural linear não supervisionada, que detém uma camada única totalmente conectada, cuja saída é organizada em uma ou duas dimensões.

Apresenta treinamento não supervisionado e competitivo, durante esse processo a rede aumenta a semelhança entre o neurônio escolhido e os seus vizinhos.

Neurônio vencedor é aquele que apresenta maior nível de ativação, o desempenho pode ser medido por erro quadrático.

### ### 4.6 Deep learning

#### Deep Learning

Nível mais profundo de programação, ocupa-se em verificar as possibilidades de aprendizagem em um nível mais aprofundado, com a utilização de RN, visando aprimoramento de reconhecimento de fala.

A diferença fundamental entre o machine learning tradicional e o deep learning está no método de extração de características relevantes para a classificação. No machine learning tradicional, um programador manualmente define quais características são importantes para distinguir diferentes classes de dados. No entanto, no deep learning, as características são extraídas automaticamente pelo algoritmo. O algoritmo é exposto a um grande volume de dados e, de forma dinâmica, determina quais características são relevantes para separar as entradas em classes e quais classes devem ser consideradas. Essa abordagem automática torna o processo de aprendizado mais eficiente e permite lidar com dados complexos e de alta dimensionalidade.



Necessitam de um grande volume de dados para definição destas características

Tem como base o backpropagation

O backpropagation é um algoritmo crucial na história das redes neurais. Sem ele, o treinamento de redes de aprendizado profundo seria praticamente impossível da maneira como é feito hoje. Ele permite que o treinamento de modelos profundos seja computacionalmente viável e eficiente. As redes neurais modernas se beneficiam do backpropagation para acelerar o treinamento em comparação com abordagens ingênuas. Essa diferença é especialmente importante ao comparar modelos treinados em um curto período de tempo com modelos que exigem anos de treinamento.

Também importantes em áreas como previsão do tempo.

O DL não possui estrutura linear, mas múltiplas camadas em cascata.

São unidades compostas por RNA, e seus parâmetros são estimados durante o treinamento.

Tarefas mais complexas que o DL pode realizar: Tradução de máquina, transcrição de dados, síntese e reconhecimento de fala, entre outras.

Modelos de DL como as redes neurais convolucionais, são capazes de localizar e classificar os objetos em imagens.

Rede neural convolucional (CNN), um algoritmo de aprendizado profundo, realiza uma série de ações como a captura de imagem de entrada, a atribuição de importância aos aspectos e uma habilidade de distinguir um do outro. A ConvNet apresenta um nível de pré-processamento menor se comparado a outros algoritmos de classificação. Técnicas antigas necessitam da apresentação dos filtros de forma manual, porém a ConvNet é capaz de compreender estes filtros e suas características.

Frisando a diferença entre RN simples e DL.

O DL se caracteriza principalmente pela utilização de camadas de neurônios matemáticos. Visam executar ações para processamento de dados, compreensão de fala e identificação de objetos de maneira visual. A informação é transmitida por meio de cada camada, saindo da anterior e disponibilizando uma entrada para o uso na próxima camada.

Primeira camada = camada de entrada.

Camada oculta.

Última camada = camada de saída.

Camada é um algoritmo com características uniformes e simples, apresentando um modelo de função de ativação.

O DL é responsável pela evolução atual do processamento de linguagem natural. Também se caracteriza pela retirada de recursos. Esta retirada utiliza um algoritmo para de maneira automatizada desenvolver recursos dos dados, objetivando o aprendizado e a compreensão. Ciência dos dados é responsável por retirar esse recurso.

O aprendizado profundo é uma área de grande interesse que demonstra um rápido avanço e aparente domínio em relação às abordagens convencionais de aprendizado de máquina. Apesar disso, sua explicação pode ser complexa. Há um interesse significativo na pesquisa e divulgação de informações sobre aprendizado profundo em diversos meios de comunicação. Essa abordagem tem sido especialmente importante em áreas como medicina, onde é aplicada no diagnóstico de doenças como o câncer de pele e o autismo.

Tipos de arquitetura:

No design das camadas de entrada e saída, ocorre de maneira direta, pode haver um volume de variação no design das camadas ocultas.

Existem basicamente 3 arquiteturas:

Feedforward - Muito utilizada em CNN, estes modelos medem um conjunto de mudanças que transforma as similaridades entre os casos. As ações dos neurônios são classificadas como funções não-lineares das ações executadas na camada antecedente.

Feedback - Normalmente utilizado em redes Stacked Sparse Coding. Este modelo adota um conjunto de ciclos relacionados com seu grafo de conexão. Provável que o usuário retorne ao ponto inicial seguindo as setas, que pode apresentar uma dinâmica complexa, dificultando o treinamento. As redes recorrentes são a forma mais simples de estabelecer um modelo de dados dispostos em sequência. Apresentam capacidade de indicar informações em seu estado oculto por um período mais extenso, porém é muito complexo treiná-las para utilizar este potencial.

Bidirecional - Muito utilizada no modelo Deep Boltzmann e Autoencoders. Funcionam de maneira similar às redes recorrentes, porém, existe simetria entre as conexões e são mais fáceis de serem analisados. Algumas não apresentam unidades ocultas, são classificadas de redes Hopfield, enquanto com redes ocultas são conhecidas como máquinas de Boltzmann.

Estas arquiteturas operam em diferentes tipos de treinamentos:

Puramente supervisionado: tem inicialização de parâmetros de maneira aleatória e utiliza tipicamente o método do gradiente descendente e backpropagation;

Não supervisionado e classificação no topo: Em cada camada é realizado o treinamento não supervisionado. Depois são treinadas as camadas supervisionadas, mantendo as outras fixas;

Não supervisionado e sintonização global supervisionada: Em cada camada não supervisionada é efetuado o treino. Depois, é adicionada uma camada de classificação e retirado o supervisionamento de toda a hierarquia.

Redes neurais profundas.

Os autores tomam como base, para os estudos que provam que redes neurais profundas são melhores que as simples, Transformação rápida de Fourier (FFT), que faz uso da fatorização esparsa da matriz de transformação discreta de Fourier (DFT). A DFT é composta por duas expressões numéricas:  $n \log n$  que resulta em elementos não nulos na matriz, e  $n^2$  que resulta em elementos não nulos na matriz DFT. Na FFT é possível idealizar, na FFT o nível das operações é elevado em apenas uma contagem de  $n \log n$  para  $n^2$ .

Neurônio - Modelo que apresenta um cálculo das entradas, através da multiplicação por pesos, e esse valor está inserido em uma função de ativação.

Muito difícil dizer qual aprendizado é dito profundo.

Nos utilizamos as bibliotecas profundas: Keras, TensorFlow e PyTorch, mesmo quando construímos uma pequena rede com cinco camadas, isso porque é mais adequada do que todas as ferramentas que viemos antes e mesmo assim chamamos apenas de redes neurais.

TensorFlow é considerado como a primeira estrutura de Deep Learning, criada no ano de 2017.

Keras é uma API de nível elevado, que se localiza sobre outras estruturas pertencentes ao Deep Learning, suportando, dentre outros componentes, o TensorFlow e CNTK. O Keras é de fácil utilização e pode começar a funcionar de maneira rápida.

O PyTorch é definido como uma estrutura de Deep Learning criada por um grupo de pesquisa de Inteligência Artificial. Habilidade de abstração das partes complexas da programação realizada em redes profundas. Codificação de nível mediano entre o Keras e o TensorFlow, grande nível de flexibilidade e controle mais elevado que o Keras.

As RN apresentam algumas aplicações, como:

Deteção e classificação de objetos.

Processamento de linguagem natural;

Identificação de faces;

Segmentação semântica de imagens;

Reconhecimento automático de voz.

Também ganharam notoriedade com a descoberta de técnicas especializadas em treinamento profundo e a concretização das GPUs.

A partir daqui foi possível evoluir através da arquitetura AlexNet. Posteriormente superada pela VGG, depois a ResNet que se caracteriza por superar a habilidade humana em classificar imagens para o ImageNet, o ImageNet é uma série de dados formados por uma quantidade imensa de imagens, o que demonstra as possibilidades que as redes profundas apresentam.

Complexidade da Teoria.

Existem limitações quanto ao uso de redes neurais profundas, cuja forma de aprendizado se dá pela aplicação de uma série de transformações nos vetores de entradas. Essas transformações são feitas pelos pesos que passam por atualização durante a etapa de treinamento.

O mapa de entradas precisa ser contínuo.

("Suave" significa que as transições entre os valores de entrada devem ser graduais e sem saltos bruscos. Isso implica que pequenas alterações nos valores de entrada devem resultar em alterações proporcionais nas previsões do modelo. Se houver saltos abruptos ou descontinuidades nos dados de entrada, pode ser mais difícil para o modelo aprender e generalizar corretamente.

"Contínuo" refere-se à ideia de que o modelo deve ser capaz de lidar com uma ampla gama de valores de entrada em um intervalo contínuo. Isso significa que o modelo deve ser capaz de fazer previsões para valores intermediários entre os exemplos de treinamento fornecidos. Se houver lacunas ou falta de cobertura em certas regiões do espaço de entrada, o modelo pode ter dificuldade em fazer previsões precisas nesses pontos.)

Outra limitação é a abstração e adaptação da rede, precisam de grande número de dados categorizados, para aprender conceitos simples, diferentes dos humanos.

Divididos em dois tipos de modelos:

Redes neurais:

Convolucionais (CNN)

Recorrentes (RNN e LSTM)

Modelos probabilísticos:

Restricted Boltzmann Machine (RBM)

Rede Deep Belief Network (DBN)

Autoencoders (AE)

o Algoritmo Long Short-Term Memory (LSTM) é um modelo de RN recorrente, tem a capacidade de separar ocorrências recentes e distantes. Isto ocorre inserindo pesos distintos para cada um e apresentando a habilidade de descartar memórias irrelevantes para prever saída posterior. Bom em sequência de dados longas.

Definidos como tipo usual e profundo de redes neurais. Dada para tratar do desaparecimento do gradiente, que ocorre com redes neurais comuns quando tem grande extensão de dados. LSTM lida com série de dados, estabelecendo o nível das falhas contínuas por meio de unidades específicas denominadas gates, que permitem ajustes de medidas da mesma forma que a sequência, quando a informação é desnecessária.

Usada em processamento de linguagem natural e identificação de caligrafia. Também bastante utilizada em previsões de preços de ações. Normalmente utilizadas para prever tendências de preço, em redes recorrentes, além de adquirir resultados elevados.

Redes Convolucionais - Modelo inspirado no córtex visual e que abrange o campo da visão computacional.

Desenvolver uma CNN pode ser custosa, que demanda certo tempo para ser executada. Diante desta situação, um conjunto de APIs é criado com o objetivo de possibilitar que empresas adquiram insights de maneira imediata.

Google Cloud Vision: É uma API do Google que realiza o reconhecimento visual utilizando a estrutura TensorFlow. Possui recursos avançados como identificação de rostos, objetos e rótulos abrangentes.

IBM Watson Visual Recognition: Faz parte do Watson Developer Cloud e possui um conjunto de classes embutidas. Permite o treinamento de classes personalizadas com base em imagens fornecidas. Oferece recursos como identificação de conteúdo impróprio (NSFW) e reconhecimento óptico de caracteres (OCR).

[Clarif.ai](http://clarif.ai/): É uma API de identificação de imagens com uma abordagem flexível. Permite a adaptação do algoritmo a diferentes domínios e temas específicos. Utiliza uma API REST para realizar suas funcionalidades.

APIs são mais generalistas, porém, é possível adquirir resultados mais expressivos, criando uma solução específica para atividades personalizadas.

Aplicação em redes neurais convolucionais, adição de sons em filmes silenciosos, o sistema precisa sintetizar os sons. O sistema passa por um treinamento usando vários vídeos que

utilizam um som de bastão, alcaçando superfícies e sons distintos. O modelo de aprendizagem profunda realiza a inteligência entre frames do vídeo e os sons do banco de dados. O sistema é analisado com o auxílio de uma configuração parecida com o Teste de Turing.

Do tipo feedforward, David Hubel e Torsten Wiesel, encontraram dois tipos de células no córtex funcional do gato: células simples (S-Cell) e complexas (C-Cell). As simples rastreiam características e as complexas sintetizam dentro de uma vizinhança retinotípica, as saídas das células simples.

Necognitron - primeira rede neural com ideais neurofisiológicos, depois surge o Cresceptron, capaz de adaptar sua topologia durante seu treinamento em 2016 as primeiras redes neurais são implementadas em GPU.

Inception v3 - Rede neural convulsional, identificação de imagem utilizando de maneira ampliada e com alto índice de precisão. Formado por blocos do tipo simétrico e assimétrico.

Arquitetura:

As Convnets são organizadas em estágios, cada um composto por camadas de convolução, seguidas de uma de amostragem (pooling). Após a final, são adicionadas camadas complementares, de maneira que fiquem interconectadas.

Camada convolucional - Camada convolucional (Convolutional layer): Nesta camada, os neurônios estão conectados a uma região específica das entradas da rede, permitindo o mapeamento de todas as regiões. Isso ocorre devido ao compartilhamento de parâmetros.

Camada de amostragem (Pooling layer): Reduz a vizinhança dos dados de entrada, selecionando o valor máximo. Essa operação reduz as variações na representação do recurso. A função de max pooling não possui parâmetros de otimização e o erro é passado para a próxima camada.

Camada fully-connected (Fully-connected layer): Liga a última camada de neurônios à saída da classificação. Cada neurônio nesta camada possui pesos associados a cada vetor de entrada. Em CNNs, essa camada é chamada de FC e é posicionada após as camadas convolucionais. Ela redimensiona os dados das camadas anteriores, realizando uma combinação linear dos vetores.

As CNN ganharam popularidade quando foram usadas para o reconhecimento de objetos a partir do mapa de características extraídas de imagens de forma automatizada. Sendo assim, a CNN desempenha o papel de extrator de característica de forma autônoma.

## Algoritmos de otimização CNN

Usa-se o algoritmo de gradiente descendente em um conjunto com método de backpropagations, que permite obter a sequência de parâmetros presentes na rede utilizando a regra de cadeia.

**Gradiente Descendente Estocástico (SGD):** É um método que acelera o treinamento da rede ao utilizar amostras aleatórias dos dados em cada iteração. O SGD analisa apenas uma amostra por vez, em vez de todo o conjunto de dados.

**Momentum:** Ao adicionar o fator momentum, um novo hiperparâmetro é introduzido. Isso permite controlar a velocidade das mudanças nos parâmetros "W" da rede, ajustando o peso da direção atual do gradiente. Isso previne que a nova atualização, chamada de "W+1", se afaste muito da direção atual no espaço dos parâmetros.

Aspectos a serem considerados no treinamento de CNNs:

- **Inicialização:** Os pesos da rede são inicializados com números aleatórios, normalmente sorteados de uma distribuição gaussiana.
- **Tamanho do minibatch:** É necessário definir o tamanho do minibatch de exemplos que será utilizado durante o treinamento com o método SGD e suas variações.
- **Regularização:** Para lidar com problemas de múltiplos parâmetros e encontrar boas soluções, é comum utilizar a norma L2 para regularizar os pesos da rede.
- **Dropout:** É uma técnica utilizada para reduzir o overfitting durante o treinamento, aplicando uma probabilidade de desativação dos neurônios.
- **Batch normalization (BN):** É um método de normalização que mantém a média de ativação próxima a 0 e o desvio padrão próximo a 1 em cada batch de entrada.
- **Data-augmentation:** Consiste em gerar uma base de treinamento aumentada, aplicando operações de processamento em cada imagem original para gerar novas imagens.
- **Pré-processamento:** As imagens de entrada podem passar por diferentes formas de pré-processamento, como a computação da imagem média e a normalização do Z-score.
- **A capacidade da CNN de aprender características distintivas dos objetos por conta própria é um ponto forte dessa técnica, eliminando a necessidade de inserção manual dessas características.**

Resumidamente, esses aspectos envolvem a inicialização dos pesos, definição do tamanho do minibatch, regularização dos pesos, utilização de dropout, aplicação de batch normalization, data-augmentation e pré-processamento das imagens. A capacidade da CNN em aprender características por si mesma é uma vantagem significativa.

## Redes neurais Recorrentes

Possuem a persistencia de informação. Utilizam eventos passados para recalibrar as saidas em relação as novas entradas. Adequadas para processamento de informação em sequencia. Dado que são capazes de memorizar eventos mesmo que eles ocorram após longos intervalos.

RNNs tem sido utilizadas em modelagem de linguagem, tradução, reconhecimento de voz. Sendo em vista que as RNNs são máquinas de Turing completas, podem desenvolver qualquer computação, como aplicação tem também a geração de legendas de imagens.

RNNs são o segundo tipo mais popular para reconhecimento de voz em assistentes inteligentes. utilizadas em dados sequencias, como voz, texto ou musica. Alexa, Assistente da google utilizam-nas. Capaz de prever palavras claramente e até inserem os acentos adequados. Porque os novos assistentes são treinados para falar não letras, mas frases inteiras. Podemos selecionar um monte de textos sonoros e treinar uma rede neural para gerar uma sequencia de audio mais proxima possivel do discurso original.

## Modelos de Redes Neurais e Modelos Probabilísticos

Caso especial é a máquina de Boltzmann, não temos conexões entre neurônios de mesma camada. Portanto um modelo de duas camadas sendo uma oculta e uma visível, e de sinapse não direcional. Utilização mais importante é a de aprendizagem para se construir uma Deep Belief Network.

Camada oculta são neurônios do tipo binários [estocásticos] (<http://xn--estocsticos-o7a.na/>) camada visível, são entradas binárias ligadas por Feature Detectors binários, que utilizam pesos de conexão.

## Deep Belief Network (DBN)

Modelo generativo probabilístico, diversas camadas variáveis latentes estocásticas, objetivo aprender níveis abstratos da distribuição dos dados de entrada. Algoritmos aplicados na resolução deste tipo de problema:

- Wake — Sleep: Nesse algoritmo, cada vetor de pesos reconhecido pelo neurônio X da arquitetura é utilizado na programação bottom-up para selecionar os estados binários das unidades ocultas.
- Greedily Layer-Wise Training: Os DBN são treinados de forma incremental, um a um. Cada variável latente é tratada separadamente e os pesos de saída de cada camada são utilizados para inicializar a próxima.
- Up-Down: Utiliza o algoritmo Greedily Training para aprender os pesos da DBN, mas recalibra os pesos com o método "up-down" para garantir alta qualidade nos valores gerados.



Autoencoder:

Stacked Autocoders, ou autocodificadores ou redes autoassociativas, uma estrutura que tem base no módulo RBM. Pode ser treinada de maneira não supervisionada. usados na redução de dimensões não lineares de dados.

Uma rede treinada para copiar a sua entrada para sua saída internamente, possui uma camada oculta que descreve as proteções presentes à entrada. São restritos a copiar somente entradas que assemelham com os dados de treinamento. como o Modelo é formado a priorizar aspectos da entrada de ser copiados, geralmente aprendem propriedades úteis.

Toma uma imagem como entrada e tenta reconstruí-la usando um número menor de bits do gargalo. A imagem é majoritariamente comprimida no gargalo. Compactação obtida por meio de treinamento da rede por um período e, à medida que ela aprende, tenta representar melhor a imagem de entrada no gargalo.

Possui uma pequena camada interna de neurônios que divide a rede em duas partes simétricas, sua estrutura contém blocos de TBM que diminuem a quantidade de neurônios em cada camada até a central. Sendo essa a de menor dimensão de toda a rede.

É possível treinar qualquer arquitetura de autoencoder escolhendo a dimensão do código e a capacidade do codificador e do decodificador com base na integridade do conjunto. Em vez de limitar a capacidade do modelo mantendo o decodificador superficial e o tamanho do código pequeno os autoencoders usam uma função que incentiva o modelo a ter outras propriedades além da capacidade de copiar sua entrada para a saída esperada.

Mais utilizadas:

- Encoder: Essa parte da rede reduz a entrada em um número menor de bits, comprimindo-a. O espaço representado por esses bits compactados é chamado de "espaço latente" ou "gargalo". O gargalo é o ponto máximo de compressão, onde a entrada é comprimida ao máximo. A codificação resultante dos bits compactados é chamada de "codificação de entrada".

- Decodificador: Essa parte da rede tenta reconstruir a entrada utilizando apenas a codificação de entrada. Quando o decodificador é capaz de reconstruir a entrada de forma exata, podemos dizer que o codificador produz as melhores codificações.

![[Untitled]](<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c358400a-4b40-4dd2-9acf-4a47c99beb91/Untitled.png>)

#### ### 4.7 Aprendizagem Bayesiana, Árvores de Decisão e Algoritmos Evolucionários

Aprendizagem bayesiana

Vertente da inteligência artificial, chamada Inteligência Artificial Probabilística ou aprendizagem Bayesiana, aplicada em vários contextos desde a área da saúde até áreas vinculadas ao entretenimento.

O Machine Learning pode ser visualizado como uma ferramenta de análise de dados que promove a automatização de mecanismos de construção de modelos analíticos.

Estatística bayesiana pode ser figurada como um mecanismo estatístico no qual, por meio do teorema de Bayes, é possível combinar informações subjetivas com informações provenientes de dados observados, de modo que seja possível atualizar as probabilidades de variáveis cujo estado não é observado dado uma série de novas observações. O raciocínio lógico tem habilidade de retirar conclusões seguindo uma base ponderada do conhecimento prévio a respeito de um determinado problema. Já o probabilístico, apresenta-se adequado para situações na qual não se obtém previamente todo conhecimento a cerca de um determinado problema, o mesmo possibilita a representação de inferências prováveis, porém incertas. Podendo ser empregada para diversos fins: decisão, classificação, análise de dependência, entre outras.

Representação da Aprendizagem Bayesiana:

Mecanismo para obter representação interna da rede, tendo o poder de definir um dado domínio com o intuito de facilitar a extração do conhecimento.

Pode ser dividido em duas seções e ambas podem ser aprendidas a partir de um método especialista ou indutivo, sendo elas: aprendizagem da estrutura (e relações entre as variáveis) e aprendizagem dos parâmetros numéricos (distribuição de probabilidade). Na metodologia especialista, o conhecimento necessário é passado por um responsável o mesmo deve ser especialista.

Na indutiva, emprega com base um banco de dados de exemplos e, partindo dele, uma rede é desenvolvida automaticamente.

O aprendizado Bayesiano também pode ser caracterizado como sendo do tipo supervisionado, visto que são fornecidas ao algoritmo instâncias juntamente com suas classes.

Vantagens: possibilidade de embutir nas probabilidades calculadas o conhecimento de domínio já existente, capacidade de classificação feita pelo algoritmo serem baseadas em evidências fornecidas, que podem aumentar ou diminuir as probabilidades das classes a serem observadas em uma nova instância que se quer classificar, fornecimento de algoritmos práticos de aprendizagem e fornecimento de uma estrutura conceitual.

Desvantagens: requerimento de um conhecimento inicial de várias probabilidades, visto que um elevado número de probabilidades deve ser calculado. Alto custo para cálculo das probabilidades, o uso do classificador `naive_Bayes` se mostra como solução de redução de custos.

Calcula-se a probabilidade de  $P(C|V)$ , C a classe analisada e V o conjunto de variáveis que descrevem os padrões.

O modelo apresentado demonstra uma exemplificação particular de uma rede Bayesiana, no qual são consideradas variáveis condicionalmente independentes entre si.

O pensamento Bayesiano tem a função de providenciar uma suposição de que as quantidades de interesse são reguladas por distribuição de probabilidade. Aplicada em diversas áreas como: diagnóstico médico, aprendizagem de mapas, interpretação de linguagem, visão, dentre outras.

Aplicabilidade

Muito utilizado na filtragem de spam.

Em Python utiliza-se a biblioteca Scikit Learn, podendo ser implementado de 3 formas: Gaussian, Multinomial e Bernoulli.

- Gaussiano: É uma distribuição normal utilizada na classificação.

- Multinomial: É utilizado para contagem discreta, como a classificação de texto, onde é possível contar quantas vezes uma palavra ocorre em um documento.

- Bernoulli: É útil quando os vetores são binários. Nesse modelo, a classificação considera, por exemplo, o valor 1 para palavras que aparecem no documento e o valor 0 para palavras que não aparecem.

Árvore de decisão.

As árvores de decisão são modelos estatísticos que usam treinamento supervisionado para classificar e prever dados. Elas constroem classificadores com base nos atributos de um conjunto de dados, dividindo o problema em subproblemas mais simples. Esses modelos fornecem informações úteis sobre as classes e atributos dos dados.

Grande utilidade em mineração de dados. grande gama de aplicações: diagnóstico médico, cenários de negócios, risco de crédito, sistema de piloto automático de aeronaves.

Uma árvore de decisão é um algoritmo de aprendizado que utiliza uma variável alvo pré-definida. Ela divide uma população ou amostra em subpopulações homogêneas com base nos divisores mais significativos das variáveis de entrada.

A precisão de uma árvore de decisão é influenciada pela forma como os nós são divididos. Ao criar sub-nós, aumentamos a homogeneidade dos sub-nós resultantes. O critério para as divisões é baseado na utilidade dos atributos para a classificação. Um ganho de informação é calculado para cada atributo, e o atributo com o maior ganho é escolhido como o teste para o nó atual. Em seguida, um novo processo de divisão é iniciado. Os algoritmos utilizados para esse propósito dependem do tipo de variáveis de destino. Para classificação, os critérios de divisão mais conhecidos são entropia e índice Gini.

Existem dois tipos de árvore de decisão, a de variável categórica e de variável contínua. Para definir isso baseia-se na variável de destino.

#### Vantagens e desvantagens das árvores de decisão

As árvores de decisão possuem várias vantagens em relação a outros métodos. Uma delas é a facilidade de entendimento, pois sua representação gráfica é intuitiva e não requer conhecimento analítico ou estatístico complexo para interpretá-la. Além disso, as árvores de decisão são rápidas em identificar as variáveis mais importantes e suas relações. Isso facilita a exploração dos dados e permite criar novas variáveis que possam prever melhor a variável alvo.

Em comparação com outros métodos de modelagem, as árvores de decisão exigem menos limpeza de dados, pois são menos sensíveis a outliers e valores faltantes. Além disso, elas não têm restrições em relação aos tipos de dados, podendo lidar com variáveis numéricas e categóricas. Também são consideradas métodos não paramétricos.

No entanto, assim como em qualquer outra metodologia, o uso de árvores de decisão também apresenta desvantagens. Uma desvantagem significativa é a tendência ao overfitting, o que pode ser mitigado definindo restrições nos parâmetros do modelo e na poda da árvore. Além disso, o método não é adequado para lidar com variáveis contínuas, pois ocorre perda de informação quando as variáveis são categorizadas em diferentes categorias.

#### Algoritmos Evolucionários

Esta vertente desenvolve metodologias e mecanismos adaptivos com o intuito de empregar processos seletivos e evolutivos por cinco linhas diferentes: Algoritmos Genéticos (AG), Programação Evolutiva (PE), Estratégias de Evolução ou Evolutivas (EE), Sistemas de Classificação (SC) e Programação Genética (PG).

Destacando-se por apresentar vantagens de conseguir solucionar problemas com uma simples descrição matemática daquilo que se quer ver na solução. Podendo ser empregados na resolução de problemas, fator que confere robustez e flexibilidade em todo o processo.

#### Paradigma dos algoritmos evolutivos

Os algoritmos evolutivos são um novo paradigma na resolução de problemas, onde a garantia de obtenção da solução é substituída pela tratabilidade como uma ferramenta de propósito geral. A ideia básica desse paradigma surgiu nos anos 50, quando foi implementado em um computador o conceito de evolução natural como uma abordagem para solucionar problemas, mesmo com recursos tecnológicos limitados na época (GRACIANI, 1995).

Algoritmos Genéticos (AG): É um método de busca e otimização proposto por Holland (1962). O AG enfatiza a recombinação como principal operador de busca e utiliza a mutação com baixa probabilidade como operador secundário. Ele é inspirado nos processos genéticos de organismos biológicos em busca de soluções ótimas ou subótimas.

Programação Evolutiva (PE): Desenvolvida por Rechenberg (1973) e Schwefel (1975), a PE emprega mutações com distribuição normal para modificar vetores reais e destaca a mutação e recombinação como operadores essenciais ao processo de busca.

Estratégias Evolutivas (EE): Desenvolvida por Fogel et al. (1966), as EE destacam a mutação e não incorporam a recombinação. Elas utilizam mutações com distribuição normal em problemas de otimização de valores reais e estendem o processo evolutivo ao espaço de parâmetros.

Programação Genética (PG): Metodologia desenvolvida por Koza (1992) que aplica algoritmos genéticos a espaços de programas computacionais. Nesse método, as estruturas de dados são representadas por meio de árvores e os operadores de crossover e mutação são utilizados.