

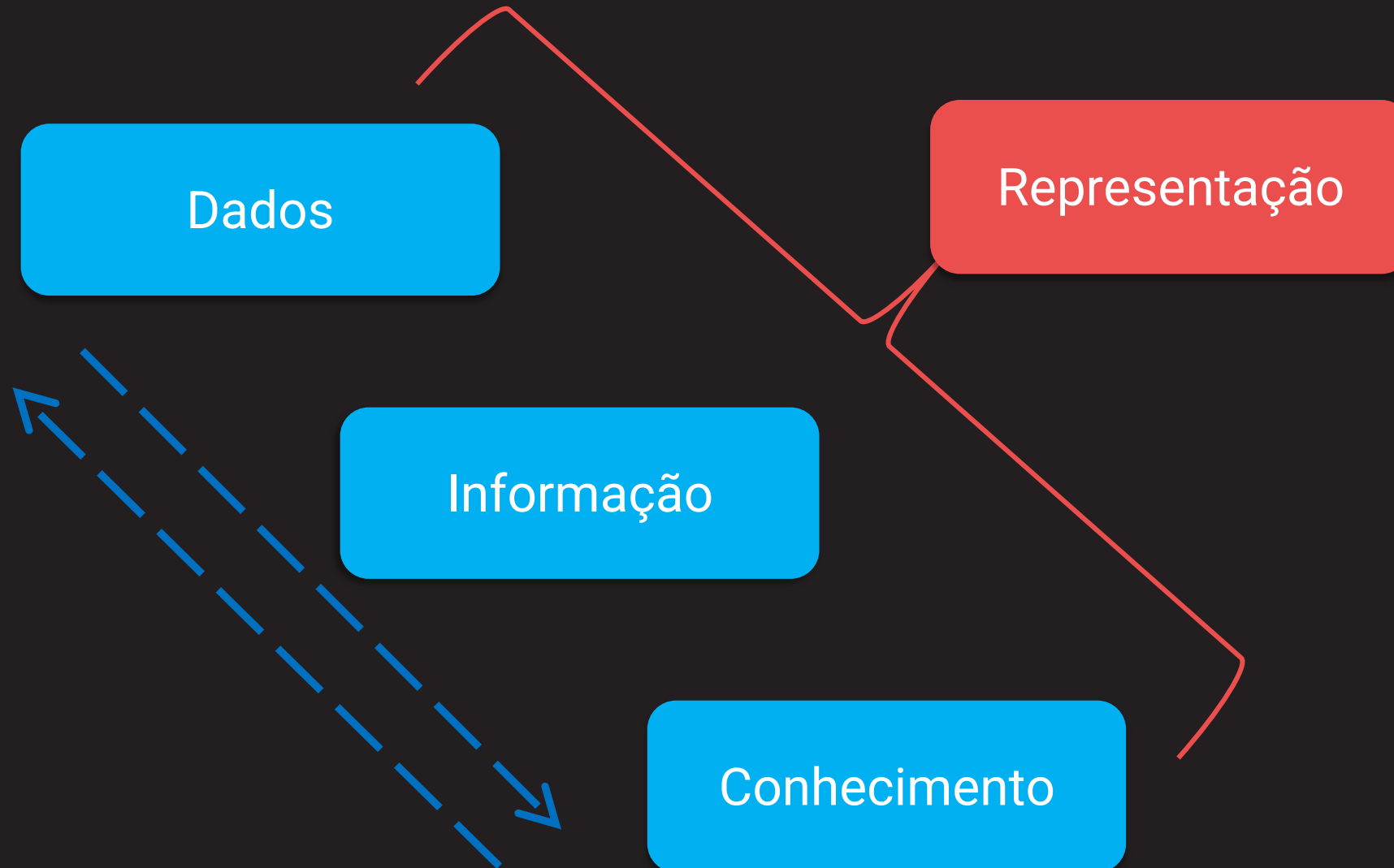
# AI & CHATBOT

Aula 12 – Representação de Dados

Prof. Érick T. Yamamoto

FIAP  
GRADUAÇÃO

# Ideia Geral



# Ideia Geral

- Dados podem ter diferentes fontes e diferentes formatos;
- A partir dos dados é extraída a informação útil gerando conhecimento;



*Leitura de instrumentos musicais*

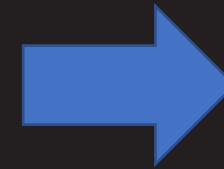


*Processos Industriais*

*Setor da Engenharia*



*Dados diversos*



*Dados brutos*



*Sensores, transdutores,*



*Imagens armazenadas*

# Representando informação

- Informação pode estar em diferentes mídias!
- A informação pode estar estruturada, semi-estruturada ou não estruturada!

TABELA

Entrada	$x_1$	$x_2$	...	$x_n$	$y$	$\hat{y}$
1	70.52	30	...	0.584	90	100
2	60.96	27	...	1.254	81	90
...	...	...	...	...	...	...
$k$	97.48	35	...	0.758	122	120

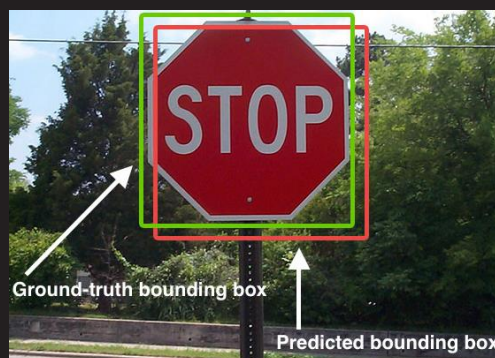
TEXTO

Esse campo de pesquisa ganhou muita notoriedade em 1986, quando David E. Rumelhart e James L. McClelland publicaram um livro que apresentou um modelo matemático computacional capaz de realizar um treinamento supervisionado dos neurônios artificiais. Esse algoritmo é chamado de **Backpropagation** e permite otimizações globais no modelo, sem restrições. Esse algoritmo também foi chamado de regra Delta generalizada, pois foi baseado na regra Delta, algoritmo de aprendizagem das redes Adalines.

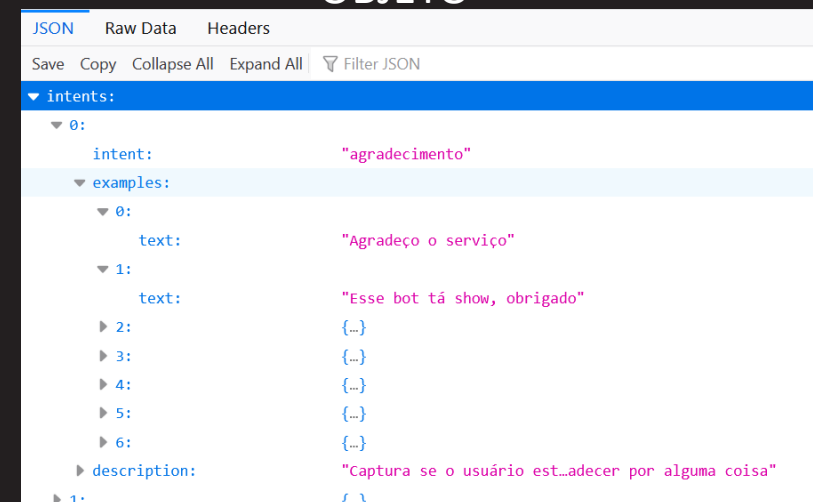
Foi a partir desses trabalhos e da criação de diversos Journals e conferências que muitas instituições fundaram institutos de pesquisas e programas educacionais que estudam redes neurais artificiais e modelos de aprendizagem.

Nos próximos tópicos vamos aprender como a rede neural pode realizar predições através do algoritmo **Feedforward** e o aprendizado ou ajustes dos pesos, com o algoritmo **Backpropagation**. Animado? Vamos começar!

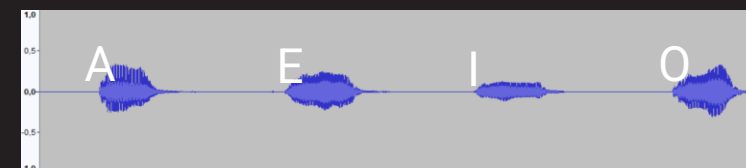
IMAGEM



OBJETO



ÁUDIO

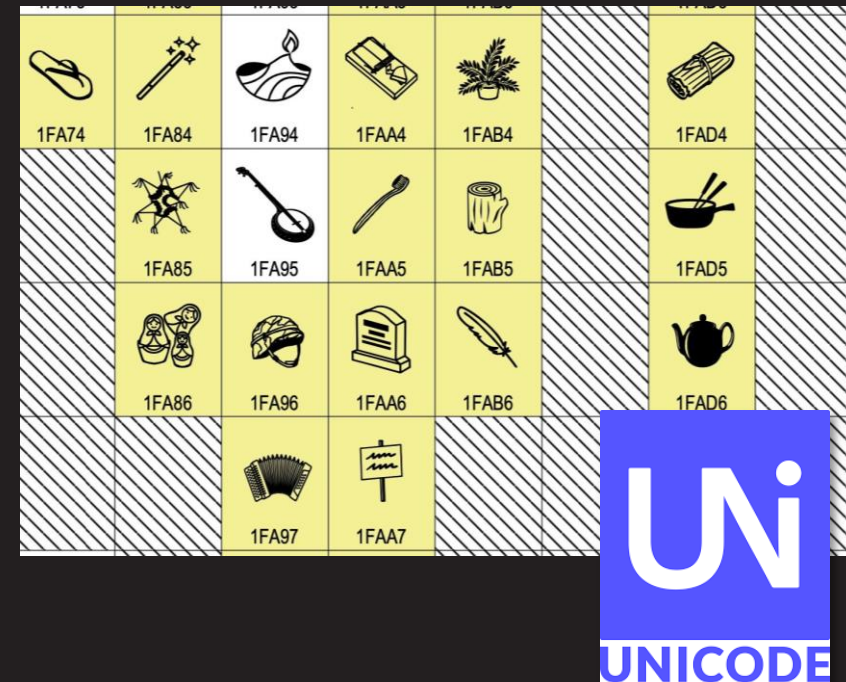


# Representando texto

- Strings são objetos em linguagem de programação usados para trabalhar com caracteres;
- Os caracteres (e mais recentemente, emojis) são **imagens mapeadas para um código hexadecimal** (e binário);
- O mapeamento hexadecimal mais conhecido é o ASCII (American Standard Code II). Para contemplar outras línguas (além do alfabeto latino) e incorporar emojis, temos o Unicode;
- Em memória, o Unicode pode ser UTF-8, UTF-16, UTF-32;

## ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



# Representando texto

- Para algoritmos de Inteligência Artificial, as strings devem ser transformadas em outras representações numéricas;
- Uma técnica muito usada na área de Processamento de Linguagem Natural é transformar as string em vetores numéricos, uma técnica chamada de **Embedding**;
- Podemos ter Word Embedding quando representamos palavras por um vetor ou ainda Sentence Embedding quando representamos sentenças por vetores numéricos;
- Existem várias formas de fazer isso, entre elas:

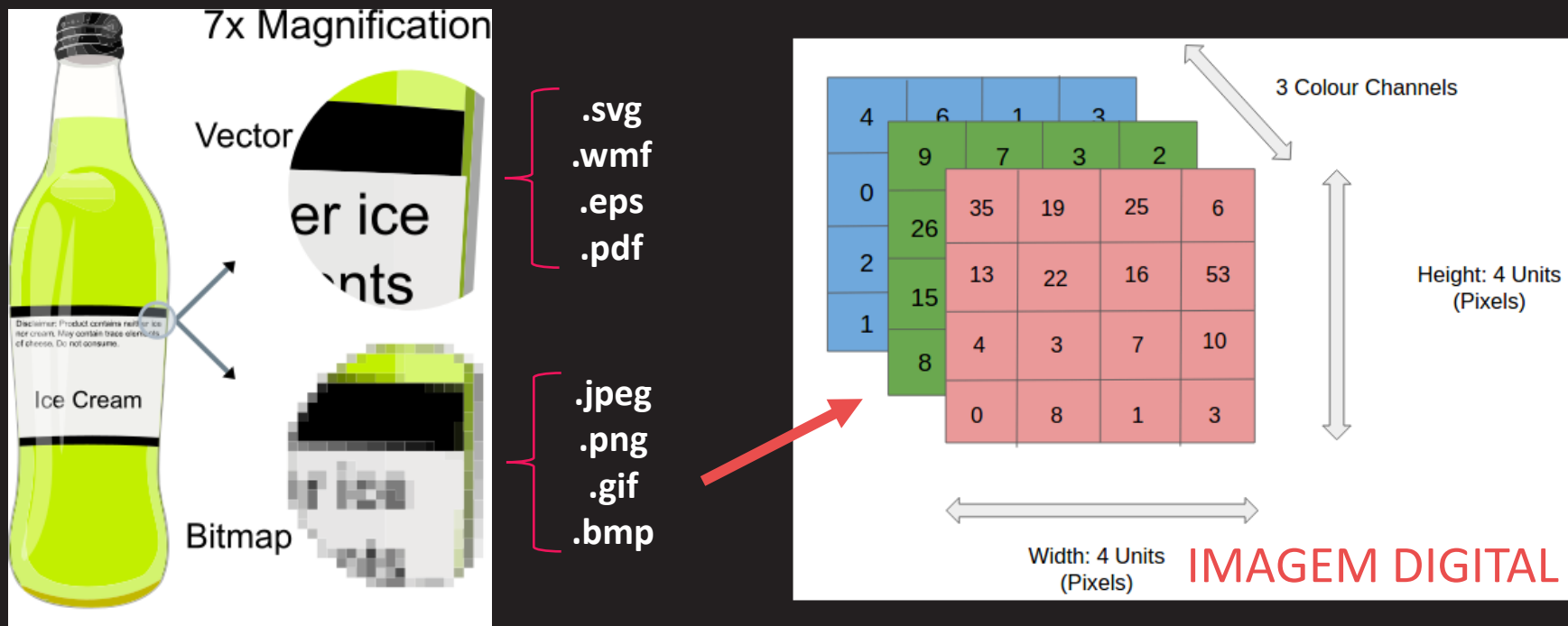
Frases: Bag of Words (BOW) ou o TF-IDF

texto = "eu vou ao cinema hoje"												
	cada	um	vou	eu	amanhã	cinema	em	hoje	e	a	ao	em
texto_vetor:	0	0	1	1	0	1	0	1	0	0	1	0



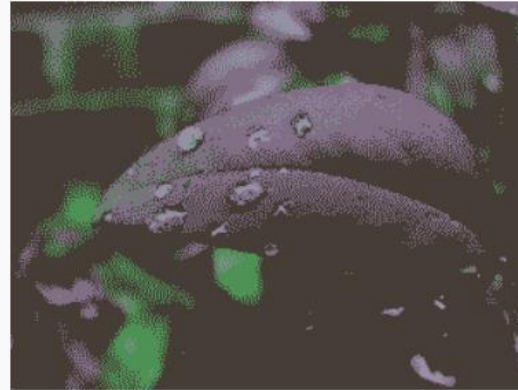
# Representando imagens

- Imagens digitais podem ter vários **formatos de codificação**;
- Imagens em formato RGB são bitmaps de 3 matrizes sobrepostas, onde cada elemento da matriz representa a intensidade daquele canal de cor naquela posição da imagem;
- **Resolução**: quantidade de pixels na altura e na largura;
- **Color depth**: quantidade de bits usados para cada número da matriz;
- Pixels próximos tendem a estar correlacionadas; já pixels distantes, não!



# Representando imagens

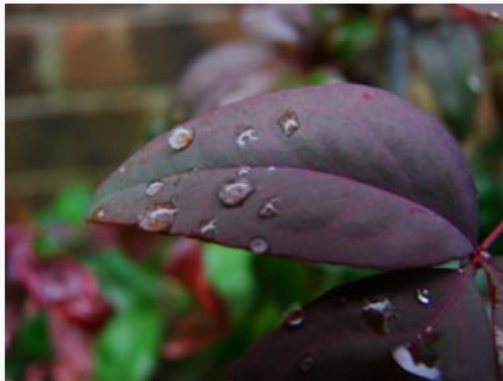
**Color depth:** como armazenamos informação digital? Quantidade usada na memória física e dinâmica? Qualidade da representação?



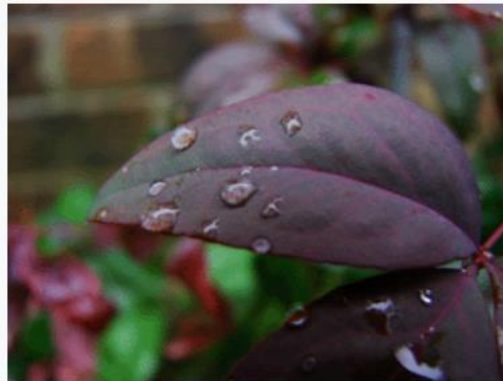
2 bit.png  
4 colors  
6 KB (-94%)



1 bit.png  
2 colors  
4 KB (-96%)



24 bit.png  
16,777,216 colors  
98 KB



8 bit.png  
256 colors  
37 KB (-62%)

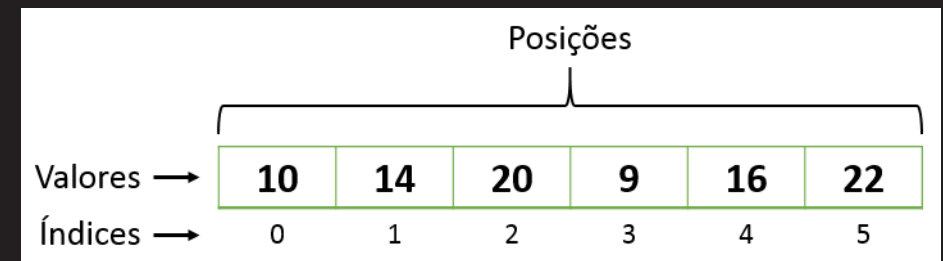
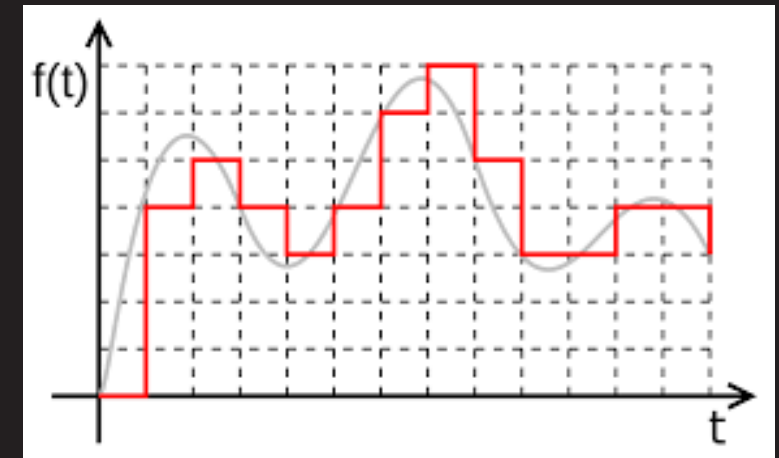
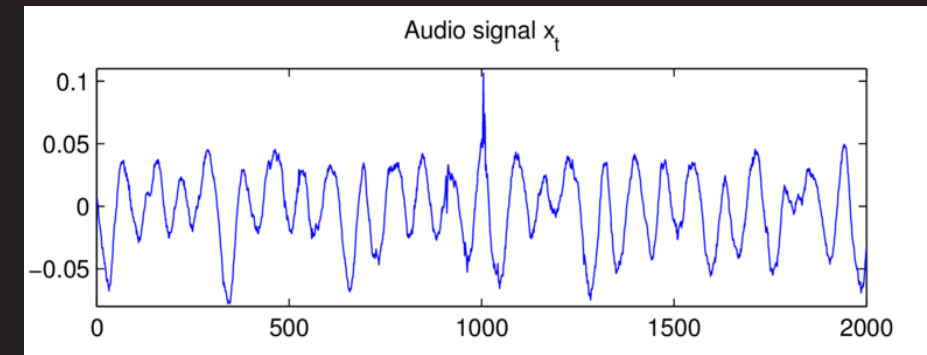
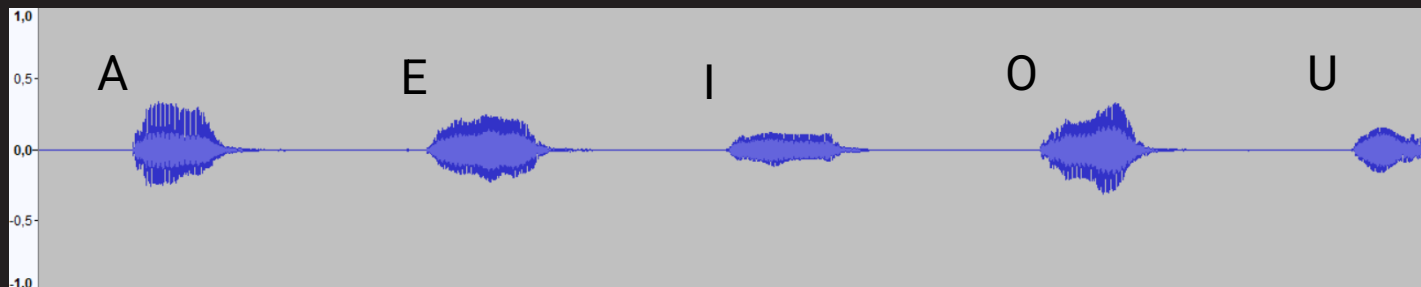


4 bit.png  
16 colors  
13 KB (-87%)



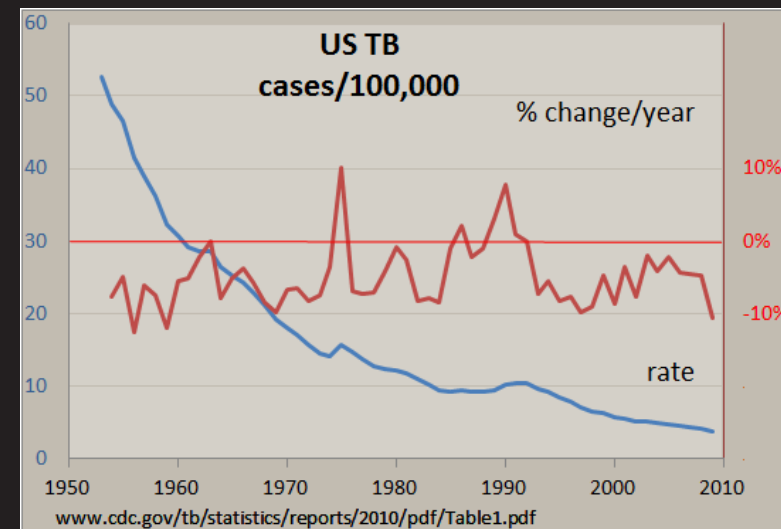
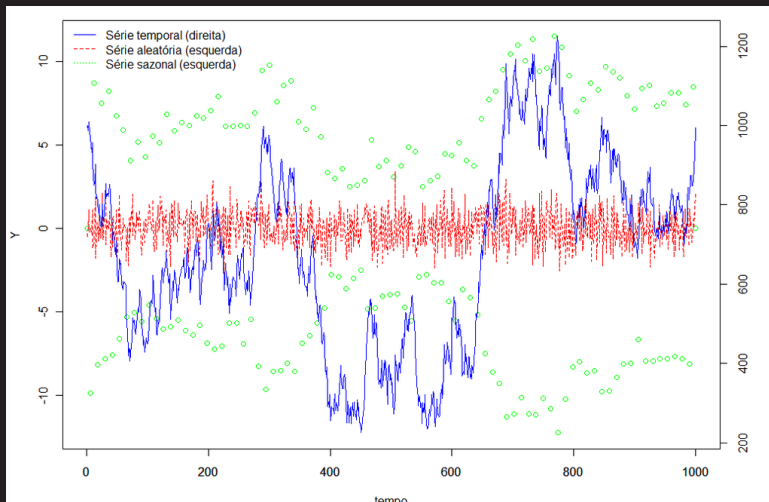
# Representando áudio

- Áudio analógico ou digital é um sinal, isto é, uma série temporal da amplitude sonora;
- No computador, áudio digital é implementado como um vetor finito, sendo o tamanho  $N$  do vetor o número de amostras de áudio, diretamente relacionado com o tempo total do som gravado;
- Além disso, áudio pode estar em um formato raw/bruto (.wav) ou em um formato comprimido (.mp3, .opus, .ogg) [codec];



# Representando “tempo”

- Além de áudio, vários outros dados podem ser representados na forma de séries temporais;
- Séries temporais associam valores a determinados pontos do tempo, e são facilmente implementador na forma de vetores (arrays ou listas);
- Outros exemplos são:
  - Valor de uma ação na bolsa;
  - Pressão do pneu em um carro;
  - Quantidade de combustível em um veículo;
  - Luminosidade em uma célula solar ao longo do dia;
  - Umidade do solo de uma plantação ao longo da semana;

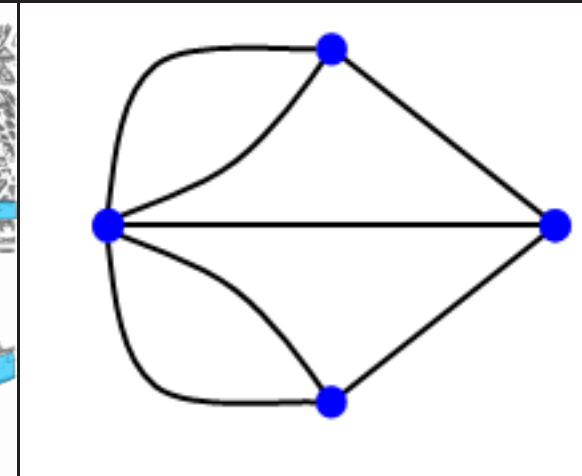
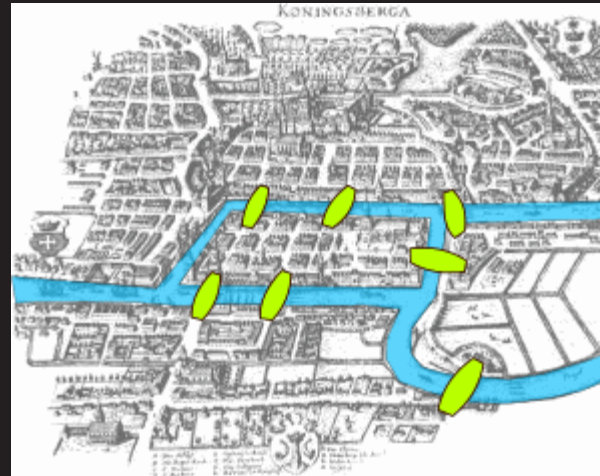
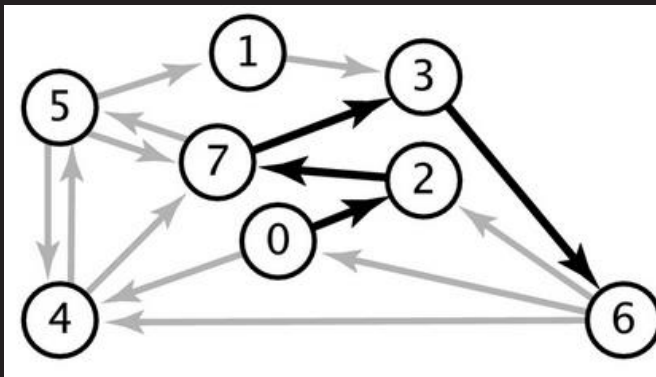


# Representação por grafos

- Grafo é um conceito matemático utilizado para **representar relação entre objetos de um mesmo conjunto**;
- Ele é amplamente utilizado na Computação para muitos propósitos: arquitetura de redes, estrutura de dados, tipos de redes neurais, sistema de arquivos, processamento de linguagem natural, busca e inteligência artificial;
- Matematicamente um grafo é um objeto denotado por  $G(V, E)$  que é composto por  $V$  vértices (nós, nodes) e  $E$  arestas (links).

Exemplo: grafo para topologia

Exemplo: grafo



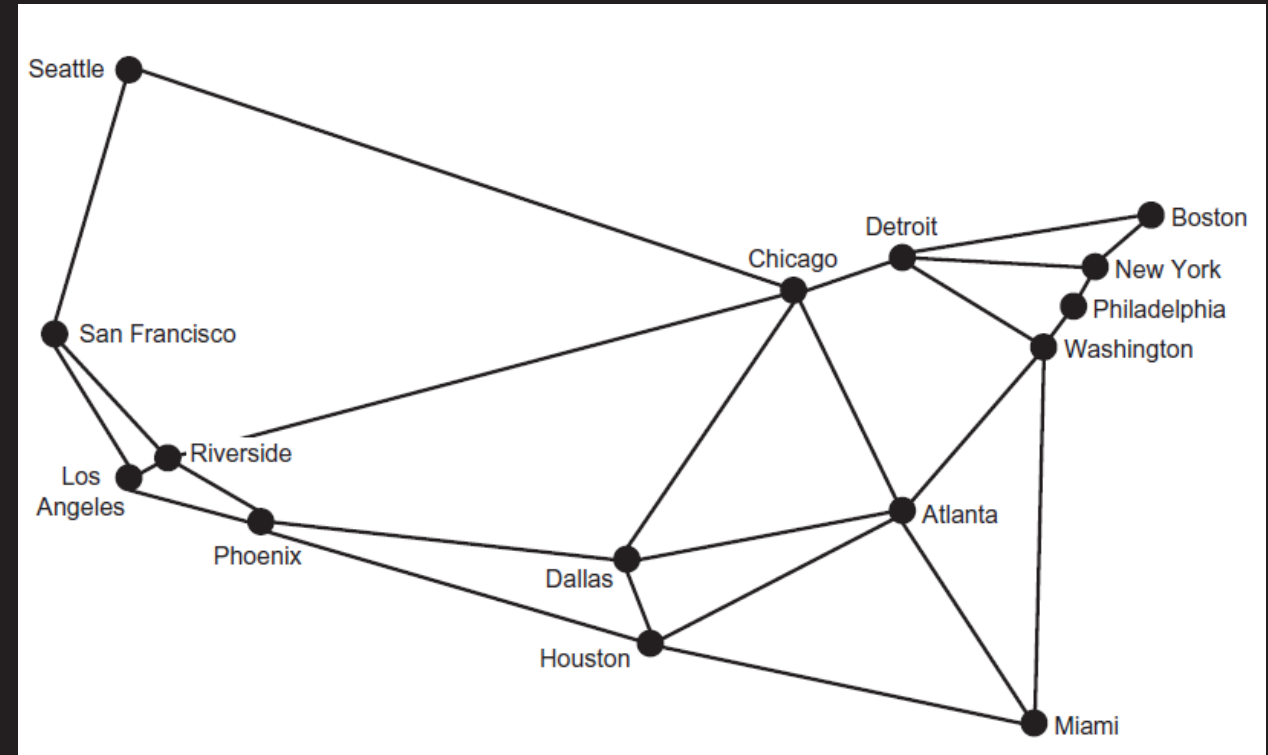
Sete pontes de Königsberg,  
Leonard Euler 1736

# Representação por grafos

- Redes sociais;
- Redes de computadores;
- Relação entre empresas;
- Relação entre países;
- Relação entre usuários do Netflix e os filmes/séries assistidos;



# Representação por grafos

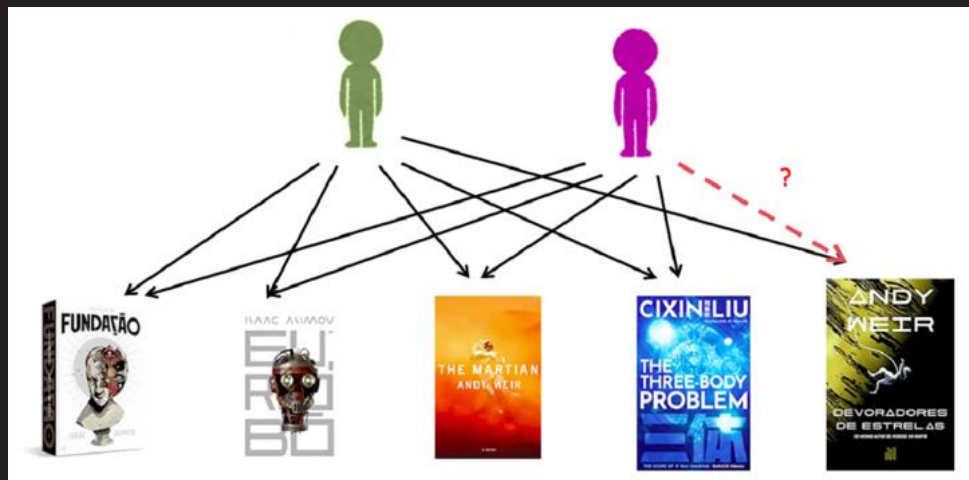
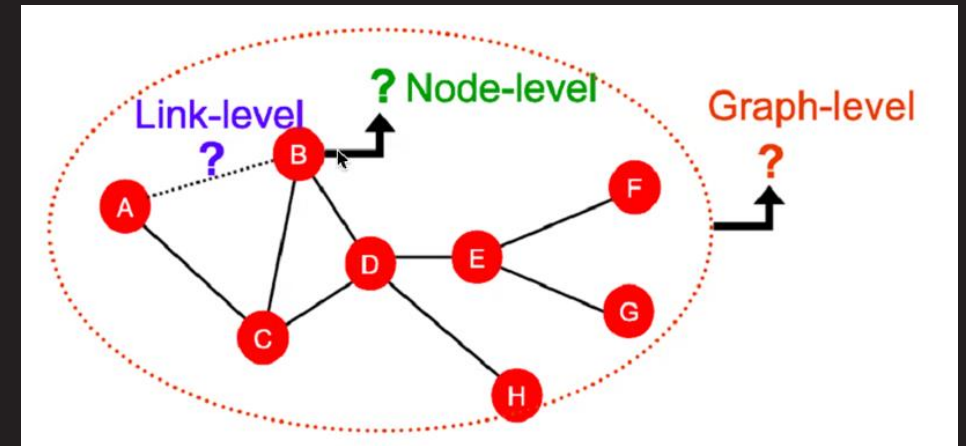


Exemplo: representação de estradas (ref. David Kopec, Classic Computer Science Problems in Python).



# Representação por grafos

- Uma vez que a informação está representada em um grafo, podemos estar interessados em tarefas como Classificação de Nós, Predição de Links, Classificação de Grafos, Otimização de Caminhos;
- Cada tipo de tarefa pode exigir um tipo de característica do grafo (feature) e um tipo de algoritmo de IA diferente;



- ❖ Busca clássica: largura, profundidade, algoritmo de Dijkstra;
- ❖ Busca heurística: A\* e Greedy Search;
- ❖ Aprendizado de Máquina: deepwalk, node2vec

Sistemas de recomendação pode ser baseados em predição de links pode exemplo;

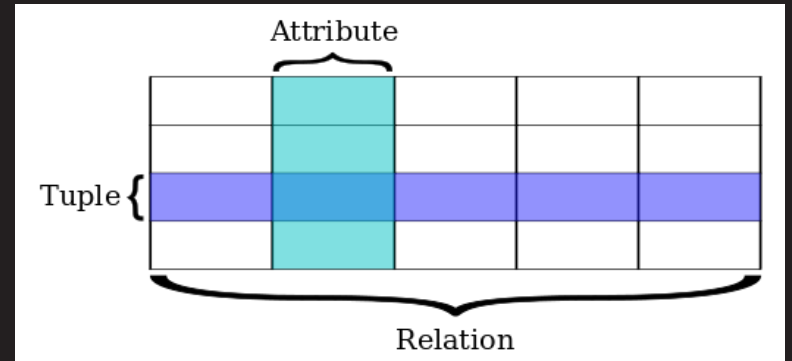
# Representação por tabelas

- A forma mais comum de representar dados é na forma de tabelas!
- As tabelas são fáceis de ler visualmente e podem ser facilmente implementadas como estruturas de dados;
- No nosso caso usaremos a estrutura de dados DataFrame da biblioteca Pandas (é uma classe);

model	engine_power	transmission	age_in_days	km	previous_owners	lat	lon	price
pop	69	manual	4474	56779	2	45.071079	7.46403	4490
lounge	69	manual	2708	160000	1	45.069679	7.70492	4500
lounge	69	automatic	3470	170000	2	45.514599	9.28434	4500
sport	69	manual	3288	132000	2	41.903221	12.49565	4700
sport	69	manual	3712	124490	2	45.532661	9.03892	4790

# Representação por tabelas

- As tabelas são fáceis de ler visualmente e podem ser facilmente implementadas como estruturas de dados (no nosso caso usaremos a estrutura de dados DataFrame da biblioteca Pandas);
- Formalmente uma tabela é uma matriz, que pode ser entendida como uma coleção de tuplas:



(pop, 69, manual, 4474, 56779, 2, 25.07079, 7.46403, 4490)

model	engine_power	transmission	age_in_days	km	previous_owners	lat	lon	price
pop	69	manual	4474	56779	2	45.071079	7.46403	4490
lounge	69	manual	2708	160000	1	45.069679	7.70492	4500
lounge	69	automatic	3470	170000	2	45.514599	9.28434	4500
sport	69	manual	3288	132000	2	41.903221	12.49565	4700
sport	69	manual	3712	124490	2	45.532661	9.03892	4790

# Representação por tabelas

- Cada linha (tupla de atributos) representa um exemplo, entrada ou instância do nossos dados;
- Exemplo: cada linha representa um carro distinto:

model	engine_power	transmission	age_in_days	km	previous_owners	lat	lon	price
top	69	manual	4474	56779	2	45.071079	7.46403	4490
lounge	69	manual	2708	160000	1	45.069679	7.70492	4500
lounge	69	automatic	3470	170000	2	45.514599	9.28434	4500
sport	69	manual	3288	132000	2	41.903221	12.49565	4700
sport	69	manual	3712	124490	2	45.532661	9.03892	4790

- Exemplo: cada coluna representa uma característica (atributo, feature) diferente;

model	engine_power	transmission	age_in_days	km	previous_owners	lat	lon	price
top	69	manual	4474	56779	2	45.071079	7.46403	4490
lounge	69	manual	2708	160000	1	45.069679	7.70492	4500
lounge	69	automatic	3470	170000	2	45.514599	9.28434	4500
sport	69	manual	3288	132000	2	41.903221	12.49565	4700
sport	69	manual	3712	124490	2	45.532661	9.03892	4790

# Exemplo de aplicações:

- Dados, informação e conhecimento são elementos fundamentalmente importantes para a área de inteligência artificial e computacional;
- Cada algoritmo de IA irá necessitar de dados em um determinado formato, isto é, cada algoritmo é feito para trabalhar com certos tipos de dados (certa representação de informação ou conhecimento);

## Exemplos:

1. Usar Rede Neurais Convolucionais para classificar imagens: a imagem precisa ter certo tamanho  $N \times M$  pixels, com certa profundidade (bits por pixel) e quantidade de canais (monocromática, RGB, CMYK, YUV).
2. Um Sistema Especialista para Diagnóstico Médico precisa de um dicionário de regras e uma ontologia, ou seja, precisa de uma representação de conhecimento que é feita em conjunto entre programadores e médicos.
3. Um algoritmo Pathfinder para definir rotas entre duas cidades usa uma representação em grafo ponderado (distância) das cidades (nós) e estradas (arestas).



# Próximos Passos

O que veremos na próxima aula

# Nas próxima aulas...

- Introdução à estatística;
- Visualização de dados;
- Introdução ao Aprendizado de Máquina;



**Copyright © 2023**

**Slides do Prof. Henrique Ferreira e adaptados pelo Prof.  
Érick T. Yamamoto- FIAP**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).