

# Inteligência Artificial Aplicada à Física

## Fundamentos e Aplicações com Redes Neurais

AULA 4

Alexandre Suaide

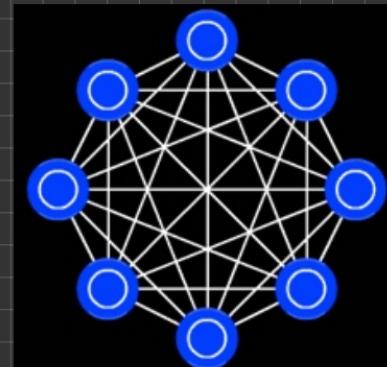
Escola Jayme Tiomno  
IFUSP 2025

# Objetivos para hoje

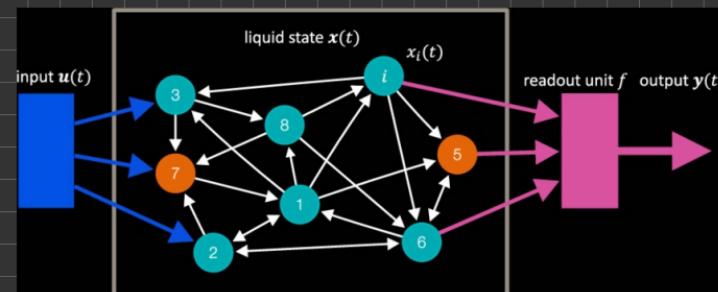
- Discutir redes neurais recorrentes
  - Redes LSTM
  - Aplicações de redes recorrentes
    - Na física  
Ex: séries temporais (clima, etc)
    - Em outras áreas  
Ex: processamento de linguagem
  - Dois exemplos práticos
    - Manchas solares
    - Análise de sentimentos em texto

Até agora

- Perceptrons e redes convolucionais
- Sempre da camada  $i \rightarrow i+1$ 
  - Feed Forward
- Há outras formas?
  - Redes recorrentes
    - Vanilla, LSTM, GRU
  - Redes em árvore
  - Feed back lateral
  - Liquid state machine



Rede de Hopfield



Liquid state NN

## Redes recorrentes

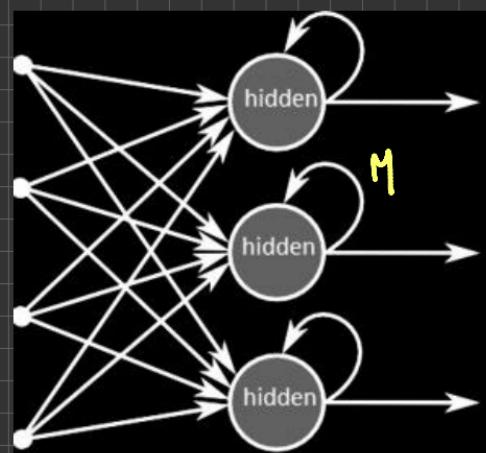
Na rede de Perceptrons

$$x^i = h[w^{i-1} x^{i-1} + b^i]$$

Nas redes recorrentes podemos pensar que o tempo (eventos anteriores) também é um fator a ser considerado. O evento anterior é usado como input no evento atual

$$x^{i,t} = h[w^{i-1} x^{i,t} + b^i + M x^{i,t-1}]$$

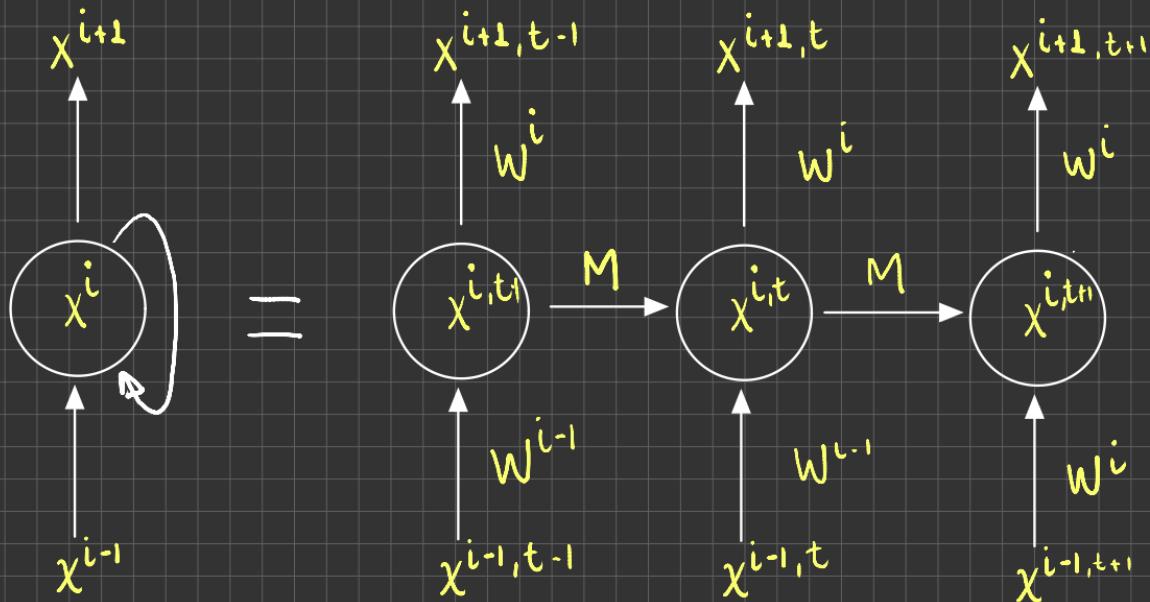
↓  
Esse é só um exemplo de uma rede recorrente clássica



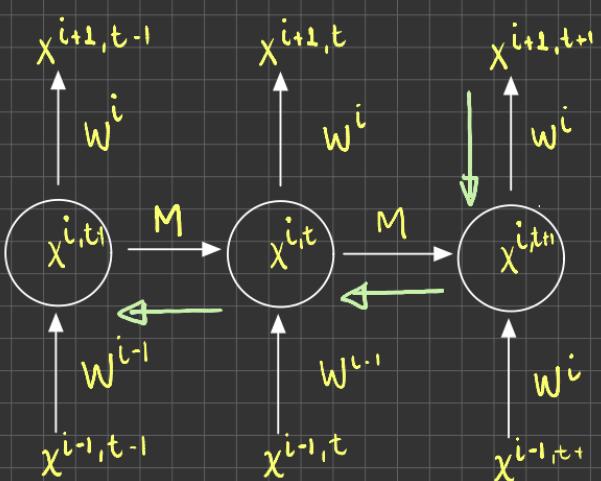
Para que servem redes recorrentes?

- O evento anterior é importante para o atual → séries
- Séries temporais → clima, mercado financeiro, ex. diferença no tempo
- Sequências estruturadas → linguagem (NLP)
  - Reconhecimento de escrita à mão ↳ Natural Language Processing
  - Reconhecimento de fala
  - Análise de sentimento de texto
  - Geração de texto
  - etc

# Treinamento de redes recorrentes Backpropagation through time



Função de custo  $\rightarrow$  MINIMIZAR EM RELAÇÃO DOS ELEMENTOS DE M



Calcular o gradiente do custo  
em relação dos elementos de M

$$\frac{\partial C}{\partial M_i} = \sum_{t=0}^{t_f} \left[ \prod_{j=t+1}^{t_f} \frac{\partial x^{i,j}}{\partial x^{i,j-1}} \right] (\dots)$$

Potencial  
Problema

← backpropagation

\* Maioria dos termos

## 1. Desaparecimento do gradiente \*

$$\left| \frac{\partial x^{i,t}}{\partial x^{i,t-1}} \right| < 1 \Rightarrow \Pi \text{ vai a zero rapidamente}$$

A rede para de aprender  $\rightarrow$  coeficientes  
não evoluem

## 2. Gradiente explosivo \*

$$\left| \frac{\partial x^{i,t}}{\partial x^{i,t-1}} \right| > 1 \Rightarrow \Pi \text{ explode}$$

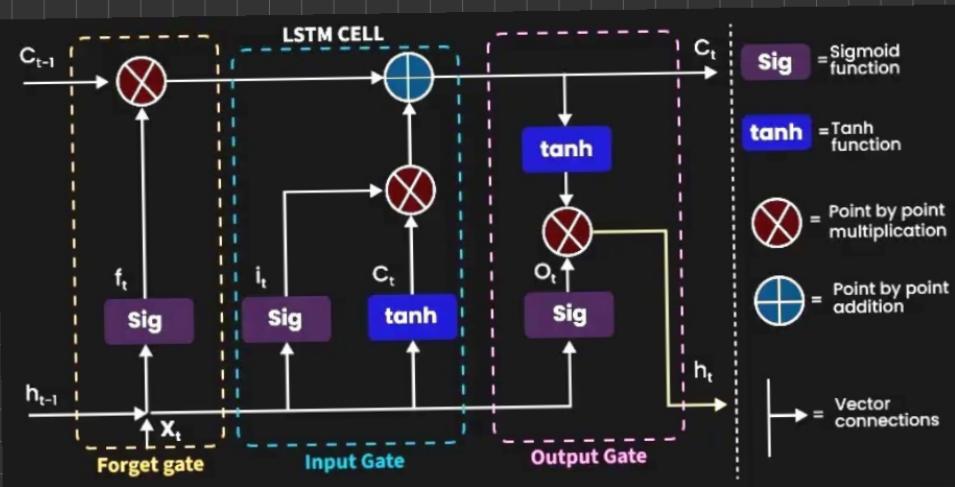
Instabilidade no aprendizado  $\rightarrow$  coeficientes  
não convergem

Lidando com esses problemas

LSTM - Long short-term memory

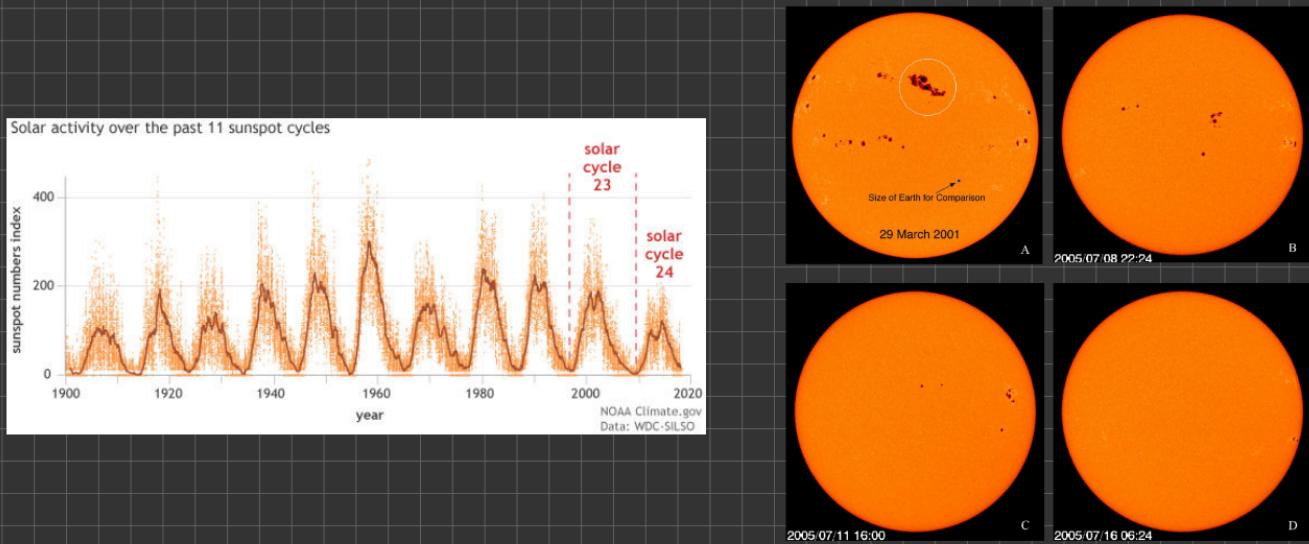
Introduz controles responsáveis pelo "esquecimento" da informação no passado

↳ Regulam o fluxo de informação no tempo que são treinados para aprender a quando lembrar ou esquecer de alguma informação



# Atividade 1 - Forecast de uma série temporal

## Previsão da ocorrência de Manchas solares



## Atividade 2 - Análise de sentimento de texto

### Como identificar sentimento?

Não deixe de assistir a  
esse filme

≠

Não assista a  
esse filme

### Como tratar texto matematicamente?

### Como tratar contexto?

CAPITAL (DINHEIRO) ≠ CAPITAL (CIDADE)

CABO (SOLDADO) ≠ CABO (FIO)

TÊNIS (CALÇADO) ≠ TÊNIS (JOGO)

⋮

Δ review	Δ sentiment
49582 unique values	2 unique values
One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. The...	positive
A wonderful little production.  ->The filming technique is very unassuming- very old-timey...	positive
I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air con...	positive
Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his par...	negative
Petter Mattei's "Love in the Time of Money" is a visually stunning film to watch. Mr. Mattei offers ...	positive
Probably my all-time favorite movie, a story of selflessness, sacrifice and dedication to a noble ca...	positive
I sure would like to see a resurrection of a up dated Seabunt series with the tech they have today i...	positive
This show was an amazing, fresh & innovative idea in the 70's when it first aired. The first 7 or 8 ...	negative
Encouraged by the positive comments about this film on here I was looking forward to watching ...	negative

Como é feito tratamento de texto?

texto → conjunto de palavras / TOKENS

TOKENS → NÚMEROS

Frase:

nginx

Copiar Editar

O gato preto dorme no sofá.

Tokens gerados:

css

Copiar Editar

[ "O", "gato", "pre", "#to", "dorme", "no", "sof", "#á", "." ]

IDs correspondentes (valores inteiros únicos do vocabulário):

yaml

Copiar Editar

[ 106, 3614, 7819, 292, 15587, 121, 9372, 298, 119 ]

- preto foi dividido em "pre" + "#to".
- sofá virou "sof" + "#á".
- O ## indica que o pedaço é um **sufixo** de uma palavra.

Como fazer correspondência por semelhança?

PARIS, LISBOA → CIDADES, CAPITALS

MACARRÃO, SUSHI → COMIDA

MACARRÃO, PIZZA → COMIDA, ITÁLIA

Semelhança depende de contexto

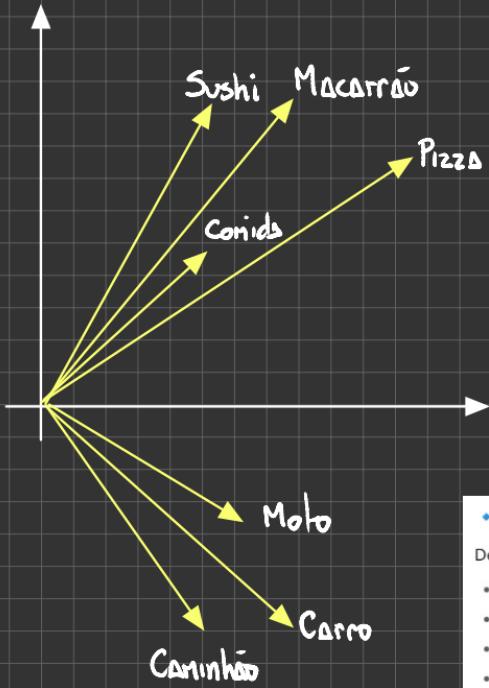
Aquela mangueira dá ótimos frutos.

Aquela mangueira está entupida.

"Attention is All you need" → ARXIV: 1706.03762

Palavras / tokens são tratados como vetores

↳ Distância pode ser medida de semelhança



## tratamento vetorial de palavras

homem - mulher ~ rei - rainha

PIZZA ~ COMIDA + ITÁLIA

SUSHI ~ COMIDA + JAPÃO

Esses vetores são determinados durante treinamento da rede

- Dimensões típicas de embeddings de tokens

Depende do tamanho do modelo. Alguns exemplos reais:

- BERT-base (português ou inglês) → cada token é um vetor de **768** dimensões.
- BERT-large → **1024** dimensões.
- GPT-2 small (117M) → **768** dimensões.
- GPT-2 medium (345M) → **1024** dimensões.
- GPT-2 XL (1.5B) → **1600** dimensões.
- GPT-3 (175B) → **12 288** dimensões.
- GPT-4 e modelos recentes → valores não divulgados oficialmente, mas estimativas indicam na faixa de **6 000–12 000** dimensões por token.
- LLaMA 2 (7B) → **4096** dimensões.
- LLaMA 2 (70B) → **8192** dimensões.

Bert-base → vetor de 768 dimensões

Frase:

```
nginx
```

Copiar Editar

O gato preto dorme no sofá.

Tokens gerados:

```
css
```

Copiar Editar

```
["0", "gato", "pre", "##to", "dorme", "no", "sof", "##á", "."]
```

IDs correspondentes (valores inteiros únicos do vocabulário):

```
yaml
```

Copiar Editar

```
[106, 3614, 7819, 292, 15587, 121, 9372, 298, 119]
```

- preto foi dividido em "pre" + "##to".
- sofá virou "sof" + "##á".
- O ## indica que o pedaço é um **sufixo** de uma palavra.

"SOF"



```
[ 0.1234, -0.0876, 0.4521, 0.0056, -0.3345,  
 0.2278, 0.0189, -0.2901, 0.5432, -0.1123, ... ]
```

Exemplo: banco de dados de review de FILMES IMDB

KERAS → textvectorization → TOKENIZA e CODIFICA O TEXTO,  
GERANDO DICIONÁRIOS

Embedding

→ VECTORIZA OS TOKENS EM UM  
ESPAÇO MULTIDIMENSIONAL



Valores VÃO ser ATRIBUÍDOS DURANTE  
TREINAMENTO

→ Custa tempo!