

# Análise de Séries Temporais com Deep Learning

---

Advanced Institute for Artificial Intelligence – AI2

<https://advancedinstitute.ai>

## Agenda

- ☐ Séries Temporais
- ☐ Conceitos
- ☐ Ferramentas para Manipular Datas
- ☐ Exploracao Estatística
- ☐ Conceitos

## Séries Temporais:

- ❑ Conjunto de observações realizadas sequencialmente ao longo do tempo.
- ❑ Muitos problemas são modelados usando séries temporais
  - Finanças, marketing, ciências sociais, entre outras
- ❑ Informações históricas, em muitos casos, auxiliam o entendimento e permite previsões

Alguns conceitos relacionados com Séries Temporais:

- ❑ Séries Univariadas:

- Um único valor ao longo do tempo

- ❑ Séries Multivariadas:

- Dois ou mais valores variam ao longo do tempo

# Análise de Séries Temporais com Deep Learning

Tempo	Característica 1	Característica 2	
1	10	10	} Passo temporal
2	20	20	
1	5	34	} Passo temporal
2	6	55	
1	3	60	} Passo temporal
2	7	90	

Característica 1	Característica 2	Característica 1	Característica 2
10	10	20	20
5	34	6	55
3	60	7	90

# Análise de Séries Temporais com Deep Learning

Tempo	Característica 1	Característica 2
1	10	10
2	20	20
3	5	34
4	6	55
5	3	60
6	7	90
1	5	5
2	8	9

Característica 1	Característica 2	Característica 1 (t+1)	Característica 2 (t+2)
10	10	20	20
20	20	5	34
5	34	6	55
6	55	3	60
3	60	7	90
5	5	8	9

Quebra da série  
para outra  
amostra

Alguns conceitos relacionados com Séries Temporais:

- ☐ Séries Estacionárias

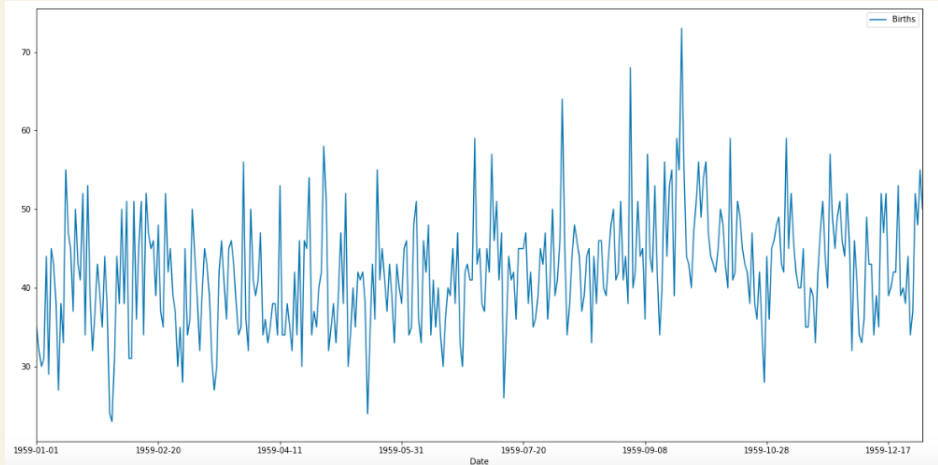
- As características apresentam padrões que não variam com o tempo

- ☐ Séries não-Estacionárias

- Padrões mudam de modo sistemático com dependência em relação ao tempo

# Análise de Séries Temporais com Deep Learning

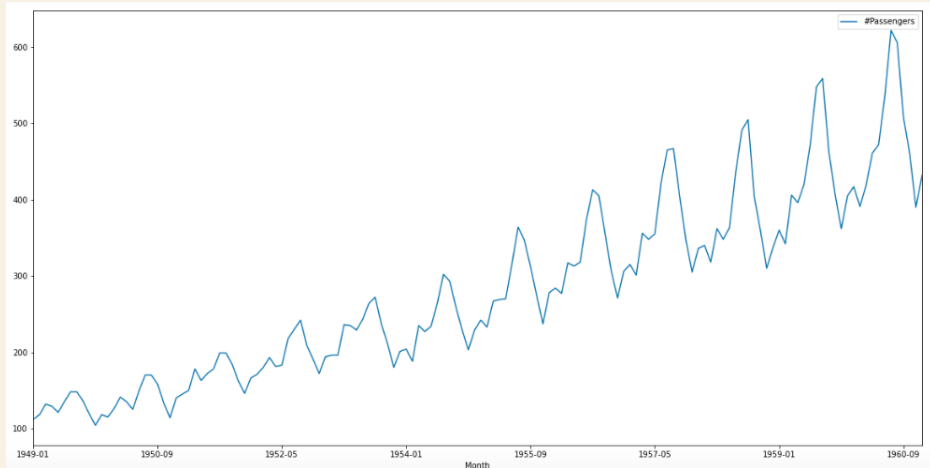
## Exemplo de série estacionária





# Análise de Séries Temporais com Deep Learning

## Exemplo de série não-estacionária



Avaliando se uma série temporal é estacionária. Métodos possíveis:

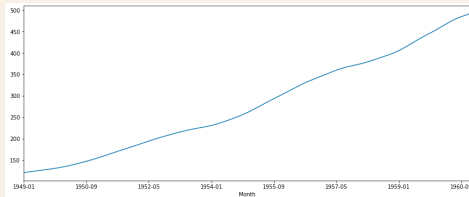
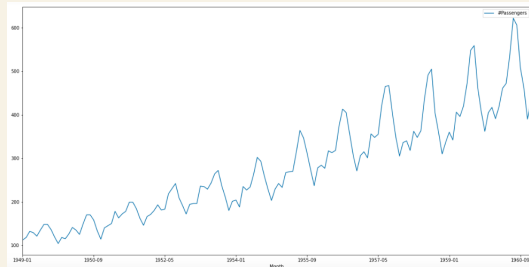
- ☐ Analisar gráficos: você pode revisar um gráfico de série temporal de seus dados e verificar visualmente se há tendências ou sazonalidades óbvias
- ☐ Estatísticas resumidas: você pode revisar as estatísticas resumidas dos seus dados para temporadas ou partições aleatórias e verificar se há diferenças óbvias ou significativas
- ☐ Testes estatísticos: você pode usar testes estatísticos para verificar se as expectativas de estacionariedade foram atendidas ou foram violadas.
  - Teste Dickey-Fuller
  - Permite identificar se uma série é estacionária, mesmo em diferentes escalas

Séries temporais não-estacionárias pode conter as seguintes características:

- ☐ Tendência: um componente linear sistemático geral ou (na maioria das vezes) componente não linear que muda com o tempo e não se repete
- ☐ Sazonalidade: ciclos que se repetem ao longo do tempo
- ☐ Ruídos : um componente não sistemático que não é Tendência / Sazonalidade nos dados

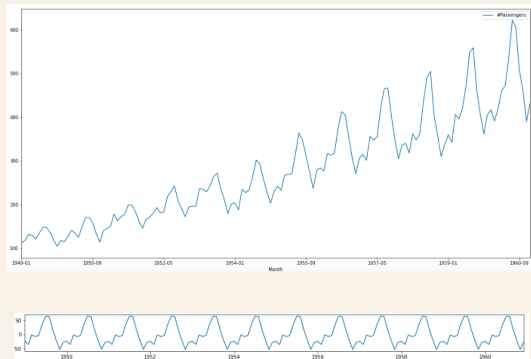
# Análise de Séries Temporais com Deep Learning

## Teste de Tendência



# Análise de Séries Temporais com Deep Learning

## Teste de Sazonalidade

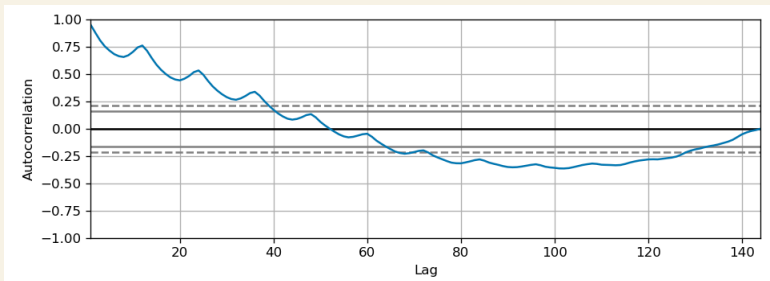


## Modelos preditivos:

- Histórico da série sendo usada para aprender o comportamento da série, permitindo prever valores para os períodos futuros
  - Para isso é importante que um conjunto específico de elementos em sequência, possua uma padrão que permita explicar os próximos elementos
  - Essa característica pode ser medida com testes de autocorrelação
- Muitos métodos estatísticos exigem que as séries sejam estacionárias para realizar tais previsões
- Alguns métodos podem ser usados para eliminar, tendências, sazonalidades e ruídos

# Análise de Séries Temporais com Deep Learning

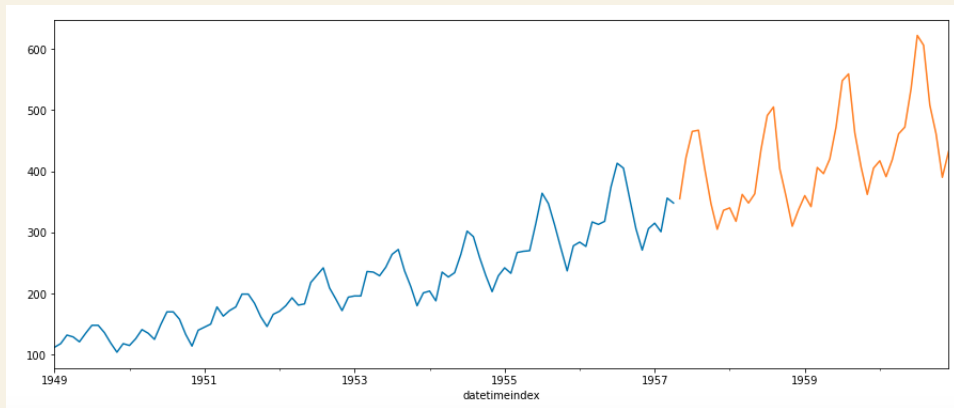
Autocorrelação Ajuda a identificar qual defasagem mínima caracteriza os próximos elementos da série



- Segmentação dos dados para treino, validação e testes
- Normalmente os dados são divididos escolhendo uma posição específica do dataset, ou aleatoriamente
- no caso de séries temporais são escolhidos períodos
  - Período de treino
  - Período de validação
  - Período de teste



# Análise de Séries Temporais com Deep Learning



Métodos de predição para séries temporais: ARIMA Auto Regressive Integrated Moving Average

- Classe de modelos que caracteriza uma série temporal com base em seus próprios valores passados (atrasos e os erros de previsão defasados) para prever valores futuros.
- Restrições:
  - Série temporal "não sazonal", sem ruídos que exibe padrões

Modelo Arima é caracterizado de acordo com 3 termos:

- ❑ AR (Auto Regressive): número de lags de  $y$  para ser usado como preditor
- ❑ MA (Moving Average): número de erros de previsão defasados
- ❑  $d$  é o número de diferenciação necessário para tornar a série temporal estacionária

A série deve ser estacionária ( ou próximo disso) para ser usada com model Arima.

- ☐ O parâmetro  $d$  define o nível de diferenciação para torar a série estacionária
- ☐ Se a série já for estacionária, basta usar  $d=0$
- ☐ Normalmente, o valor de diferenciação é definido com base no efeito na autocorrelação
- ☐ Combinando os parâmetros  $d$ , AR e MA é possível definir um modelo de regressão para estimar os próximos valores da série temporal

- ❑ Objeto datetime é disponibilizado pelo python para manipular datas
- ❑ Numpy e Pandas oferecem recursos para manipular datas
  - Slicing
  - Mudança de frequência
  - Segmentação
  - Agrupamento

- Aprendizagem de máquina são aplicados normalmente para modelar problema utilizando recursos além dos métodos estatísticos tradicionais
- É difícil prever se um algoritmo oferecerá melhor desempenho para um problema considerando que a série é estacionária ou não
- Nesse sentido, a fonte de informação ainda que não seja uma série estacionária, pode ser usada na engenharia e seleção de features

Representação de série para Deep Learning:

- A sequência é dividida em etapas fixas
  - Uma quantidade de dias, horas, semanas, elementos, etc
  - Para cada elemento da sequência um valor é associado, que é o valor da sequência
  - Esse é o conjunto de features
- O próximo item da sequência é a coluna alvo

RNN são redes neurais estruturadas de forma a representar sequências

- ❑ Para isso tais redes são montadas com base em um tipo de camada chamada camada recorrentes
- ❑ Tais camadas possuem uma unidade interna chamada memória
  - A idéia dessa unidade é processar sequências de quaisquer tamanhos



Camada recorrente:

- ☐ Camadas recorrentes recebem como entrada uma matrix multidimensional
- ☐ Cada item da sequência gera um valor de saída  $y$  e armazena uma matriz com um valor intermediário na unidade de memória
- ☐ O próximo valor é calculado usando o valor  $x$  da sequência e o valor intermediário gerado no valor anterior
- ☐ Esse laço é repetido ao longo de todos os valores da sequência

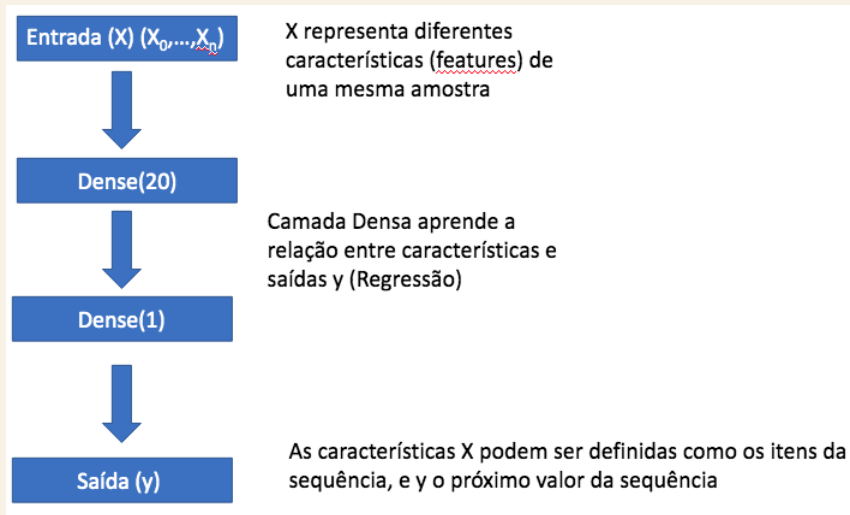
Dimensionalidade de uma camada recorrente:

- ☐ Tamanho do batch
- ☐ Passos temporais
- ☐ Dimensionalidades
  - Para séries univariadas esse valor é 1
  - Para séries com mais valores esse valor é maior

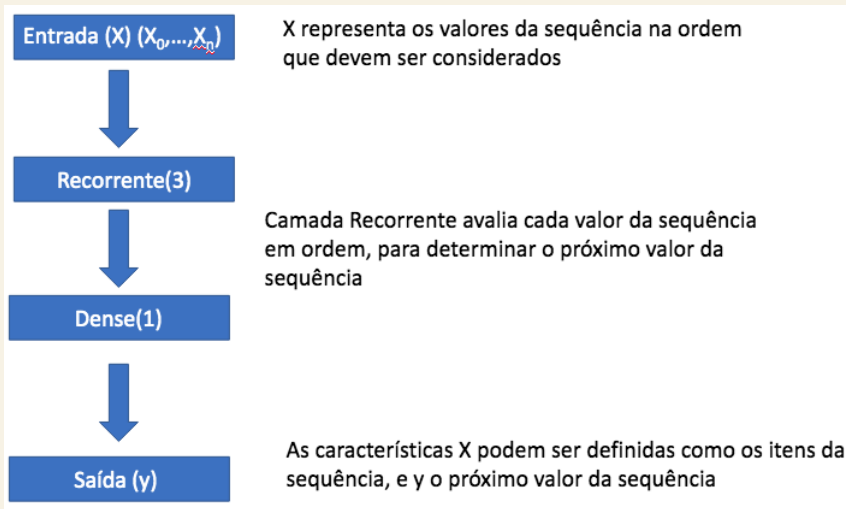
Uma rede MLP pode ser usada para prever o próximo valor em uma sequência:

- ☐ Todos os elementos da sequência são considerados como uma característica unificada do processo
- ☐ Para que a MLP possa identificar tendências, sazonalidades

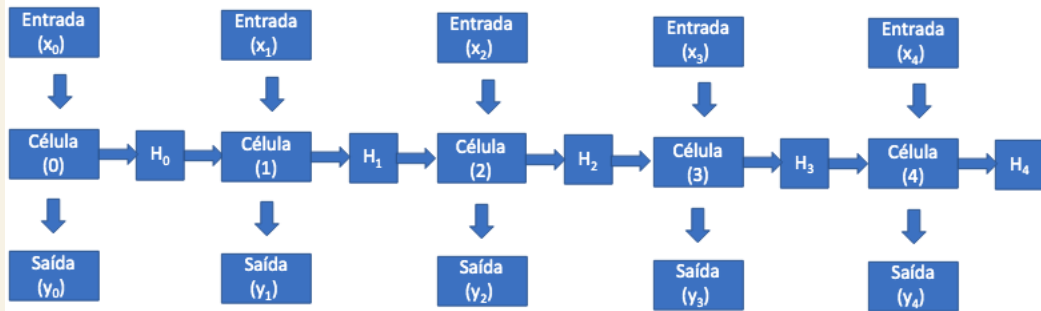
# Análise de Séries Temporais com Deep Learning



# Análise de Séries Temporais com Deep Learning



## Sequência de Tamanho 5



- As entradas são tridimensionais. Por exemplo, se tivermos um tamanho de janela de 30 registros de data e hora e os agruparmos em tamanhos de quatro, a forma será  $4 \times 30 \times 1$
- A cada registro de data e hora, a entrada da célula de memória será uma matriz  $4 \times 1$
- A célula também receberá a entrada da matriz de estados da etapa anterior.
  - No primeiro passo será zero.
  - Para os seguintes, será a saída da célula de memória.
- Além do vetor de estado, é claro que a célula produzirá um valor  $Y$

- Se a célula de memória é composta por três neurônios, a matriz de saída será  $4 \times 2$  (tamanho de batch=4 neuronios=3)
- Portanto, a saída total da camada é tridimensional, neste caso,  $4 \times 30 \times 3$ .
- Com quatro sendo o tamanho do lote, três sendo o número de unidades e 30 sendo o número de etapas gerais.



- Em uma RNN simples, a saída de estado  $H$  é apenas uma cópia da matriz de saída  $Y$ .
- Por exemplo,  $H_0$  é uma cópia de  $Y_0$ ,  $H_1$  é uma cópia de  $Y_1$  e assim por diante.
- Portanto, a cada registro de data e hora, a célula de memória obtém a entrada atual e também a saída anterior.

- ❑ Agora, em alguns casos, você pode inserir uma sequência, mas não deseja produzir e deseja apenas um único vetor para cada instância do lote. Isso geralmente é chamado de `sequence to vector`.
- ❑ Para isso basta ignorar todas as saídas, exceto a última.
- ❑ Esse comportamento é padrão do Keras é definido pela variável `return_sequences=false`
- ❑ Se deseja retornar a sequência inteira basta definir `return_sequences=true`
- ❑ Isso será necessário quando utilizar diversas camadas recorrentes empilhadas

Algoritmos de predição utilizam amostras conhecidas do problema para treino (Período de treino)

- Quando temos apenas uma única sequência de eventos é difícil identificar os dados que generalizam o problema
  - Nesse caso, é mais interessante separar segmentos da série como amostras
- Objeto dataset do keras permite montar amostras de séries a partir de uma sequência única

## LSTM

- ❑ Caso especial de RNN
- ❑ Trata o problema de longas dependências na sequência
  - Um recurso chamado Gate permite filtrar informação enviada na sequência
  - Dessa forma, a rede consegue aprender sequências com dependências mais complexas
- ❑ LSTMs Bidirecionais permitem interpretar a sequência em duas direções