

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Previsão de consumo nos estados do Brasil**

Julia Leite da Silva

MONOGRAFIA FINAL

MAC 499 — TRABALHO DE  
FORMATURA SUPERVISIONADO

Supervisor: Prof. Dr. Marcelo Finger

São Paulo  
2022

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0  
(Creative Commons Attribution 4.0 International License)*

*Dedico este trabalho a minha família,  
meu pai e minha mãe que têm iluminado  
meu dia e me dado força, alegria e paz.*



# Agradecimentos

*Fight with determination, embrace life and live it with passion. Lose your battles with class and dare to win because the world belongs to those who dare to live. Life is worth too much to be insignificant.*

— Charles Chaplin

Queria agradecer, antes de tudo, a Deus por tudo que Ele tem me dado. Uma vez minha mãe me disse que Deus abençoa o trabalho das nossas mãos, que várias vezes onde ela pensou que não daria certo ou não daria conta, ela pedia ajuda e, no final, tudo se endireitava. Ela me disse também que Deus colocava pessoas na vida dela que a ajudavam e orientavam a encontrar um caminho. Fico muito grata de enxergar isso acontecendo na minha vida também.

Agradeço aos meus pais, que têm sido minha força, alegria e lar. Agradeços aos meus amigos, amigas e namorado, por colorir meus dias e encher meu rosto de sorrisos.

Quero também agradecer ao Marcelo Finger, meu orientador e ao Felipe, que me surpreendeu com a parceria e a nessa jornada. Muito ajuda. Agradecer também ao Gabriel, pelo apoio para escrever esse trabalho.



## Resumo

Julia Leite da Silva. **Previsão de consumo nos estados do Brasil**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2022.

[illegible]

**Palavras-chave:** Palavra-chave1. Palavra-chave2. Palavra-chave3.





# Abstract

Julia Leite da Silva. . Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2022.

[illegible]

**Keywords:** Keyword1. Keyword2. Keyword3.



## Lista de Abreviaturas

IA	Inteligência Artificial
ML	<i>Machine Learning</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
FGV	Fundação Getúlio Vargas
PIB	Produto Interno Bruto
BACEN	Banco Central do Brasil
IPEA	Instituto de Pesquisa Econômica Aplicada
RMSE	<i>root-mean-square error</i>
MAPE	<i>mean absolute percentual error</i>
MAE	<i>mean absolute error</i>

## Lista de Símbolos

$\omega$	Frequência angular
$\psi$	Função de análise <i>wavelet</i>
$\Psi$	Transformada de Fourier de $\psi$

## **Lista de figuras**

## **Lista de tabelas**

## **Lista de programas**

# Sumário



# Introdução

O que prédios, pontes, hidrelétricas e aeroportos têm em comum? Todos são frutos da indústria da construção civil, um importante componente do investimento brasileiro e, consequentemente, uma das grandes engrenagens responsáveis por movimentar a atividade econômica no Brasil. Em 2021, por exemplo, segundo **cbic-report**, o Produto Interno Bruto (PIB) desse setor destacou-se com alta de 9,7% enquanto o PIB do Brasil cresceu 4,6% e o PIB do Agronegócio registrou queda de 0,2%. Dessa forma, o setor da construção figura como importante impulsionador da economia do país, capaz de gerar renda e empregos.

Nesse contexto, o cimento<sup>1</sup> é uma das principais matérias primas da indústria da construção civil. Esse material é um pó com propriedades aglomerantes que endurece quando é submetido à água, após endurecer, contudo, não é mais decomposto, mesmo em contato com a água. Além disso, ao ser misturado com areia, pedra britada, pó-de-pedra e cal resulta na argamassa e no concreto utilizados em construções. (**boletim-cimento**)

## Motivação

Contudo, a falta de um modo bem fundamentado para prever o consumo de cimento nos estados do Brasil é uma demanda entre as empresas cimenteiras. Uma nova fábrica representa para a empresa um alto investimento a longo prazo, uma vez que a construção de uma fábrica representa um alto investimento financeiro, além de tempo, por demorar anos para finalizar as obras, a exemplo da fábrica construída pela Votorantim Cimentos no Complexo Industrial e Portuário do Pecém, no Ceará, cuja construção demorou 3 anos e custou cerca de R\$ 200 milhões, conforme noticiado no portal do Instituto Brasileiro de Mineração (**fabrica-noticia**). Além disso, o cimento é um produto que não suporta longas estadias em estoque, visto que a norma brasileira recomenda o uso em até 90 dias após a fabricação, é importante, então, que a produção do material esteja fortemente alinhada ao consumo. (**abnt**)

Assim, um modelo que permitisse prever a demanda a nível de estados do Brasil poderia auxiliar gestores a tomar melhores decisões e a estruturar a estratégia de forma mais embasada, de modo a reduzir os riscos do setor. Além disso, poder-se-ia apoiar órgãos governamentais a direcionar ações para mitigar o impacto ambiental da fabricação

---

<sup>1</sup> O material conhecido na construção civil como "cimento" é denominado mundialmente como cimento de portland. Em 1824, construções com pedra de Portland eram comuns na Inglaterra, por isso, o inventor do cimento, Joseph Aspdin, ao notar que sua invenção tinha aspecto similar ao material tão difundido na época, optou por registrar a patente como "cimento de portland".

desse produto, pontuado como um grande emissor de gases de efeito-estufa em escala mundial.(**meio-ambiente**)

## Objetivos

Este trabalho, então, propõe-se a aplicar modelos de aprendizado de máquina para determinar qual é mais eficiente para prever a demanda por cimento nos estados do Brasil. Os algoritmos avaliados em ordem crescente de complexidade e robustez são: regressão linear, redes neurais *multi-layer perceptron* (MLP) e redes neurais recorrentes.



# Capítulo 1

## Fundamentação teórica

### 1.1 Inteligência artificial

Atualmente, a inteligência artificial (IA) permeia diversos momentos do cotidiano. Um exemplo é a empresa norte-americana de *streaming* Netflix, que utiliza um conjunto de técnicas de inteligência artificial para recomendar conteúdo personalizado aos usuários da plataforma de acordo com os interesses particulares de cada um. Em particular para a Netflix, não há um modelo ou algoritmo único utilizado para todas as recomendações de conteúdo, essa tarefa é dividida em subtarefas realizadas por diferentes modelos de acordo com a atividade a ser realizada e os dados disponíveis. Por exemplo, a escolha de qual vídeo será exibido para o usuário ao logar no perfil da plataforma é executada por um modelo diferente do que o que elenca os vídeos já assistidos que o membro pode continuar a ver.

Dessa forma, segundo **netflix**, a empresa proporciona uma experiência única a cada indivíduo que acessa a plataforma. Essa estratégia tem o objetivo de aumentar a satisfação a longo prazo do cliente e garantir a retenção dos membros, uma vez que a plataforma é monetizada com assinaturas mensais. **netflix** ainda ressalta que a estratégia de utilizar inteligência artificial para as recomendações tem se refletido ao longo dos anos com uma melhora na taxa de retenção dos membros.

*"Therefore, the value of a recommender system can be measured by the increase in member retention. Over years of the development of personalization and recommendation technologies, we have been able to repeatedly create meaningful improvements in retention"* (**netflix**)

Mas afinal, o que é inteligência artificial? O termo "inteligência artificial", de *artificial intelligence* em inglês, foi elaborado por John McCarthy e utilizado oficialmente pela primeira vez em 1956 no seminário de Dartmouth, um *workshop* sobre a área que reuniu os maiores estudiosos do ramo durante dois meses segundo **aima**. Entretanto, em 1950, Alan Turing já se perguntava se máquinas poderiam pensar e desenvolvia estudos e conceitos

no tema que permanecem relevantes como o Teste de Turing<sup>1</sup>. O termo "inteligência artificial, pode ser utilizado com várias conotações, uma vez que não apresenta uma definição única e aceita, conforme relembra **wang2019defining**. Uma possível definição, segundo **what-is-ai**, é a ciência e engenharia de construir máquinas inteligentes, em especial programas de computador, nesse contexto, inteligência é o aspecto computacional da habilidade de atingir os objetivos.

*"The science and engineering of making intelligent machines, especially intelligent computer programs. (...) Intelligence is the computational part of the ability to achieve goals in the world."* John McCarthy

Em 1959, o pioneiro em IA Arthur Samuel<sup>2</sup> descreveu aprendizado de máquina como o campo de estudo que dá aos computadores a habilidade de aprender sem serem especificamente programados. Aprendizado de máquina ou *machine learning* em inglês, portanto, compreende sistemas de inteligência artificial capazes de adquirir seu próprio conhecimento ao extrair padrões dos dados brutos de acordo com **Goodfellow-et-al-2016**. *Machine learning*, então, configura-se como uma sub-área de inteligência artificial.

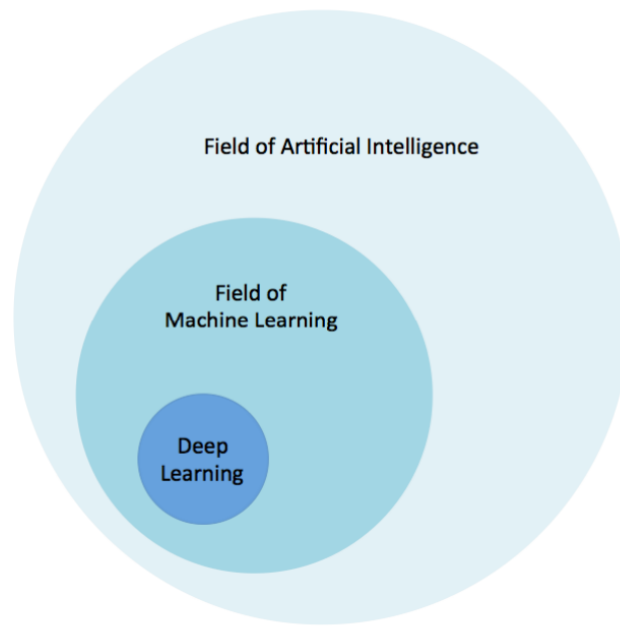
Já o aprendizado profundo, ou *deep learning*, é uma categoria específica de algoritmos de *machine learning*. De acordo com **deeplearningbook**, caracteriza-se por modelos que fazem processamento de dados com neurônios matemáticos de forma a imitar o funcionamento do cérebro humano. Nesses algoritmos, a informação é passada de camada em camada até obter a saída, trata-se, portanto, de redes neurais com várias camadas ocultas de neurônios segundo **d2l**.

A relação entre inteligência artificial, aprendizado de máquina e aprendizado profundo pode ser vista na imagem ??.

---

<sup>1</sup> Em **turing**, Alan Turing propôs o Teste de Turing com a intenção de determinar se uma máquina era inteligente. O teste de Turing é uma variação do Jogo da Imitação em que um entrevistador deve fazer perguntas a dois jogadores, um humano e uma máquina, sem qualquer distinção. Ao final, o entrevistador deve descobrir qual dos jogadores é uma máquina e qual é a pessoa. Se a máquina fosse capaz de enganar o entrevistador, seria considerada inteligente.

<sup>2</sup> Arthur Samuel foi um engenheiro e um dos pioneiros em inteligência artificial, desenvolveu um programa que jogava damas com humanos e aprendia com cada jogada dos oponentes. O programa tornava suas jogadas mais assertivas ao calcular as probabilidades de cada jogada e é considerado uma das primeiras aplicações de aprendizado de máquina.



**Figura 1.1:** Relação entre inteligência artificial, aprendizado de máquina e aprendizado profundo (dl-oreilly)

## 1.2 Tarefa

Os modelos de *machine learning* podem realizar várias categorias de tarefas, dentre elas a regressão e a classificação, a depender da atividade realizada pelo algoritmo. Neste trabalho, foram utilizados modelos de regressão onde o objetivo é prever um valor real a partir dos dados de entrada de acordo com **Goodfellow-et-al-2016**. É possível, então, descrever cada *input* dos modelos como um vetor  $x$  com  $n$  atributos (*features*) tal que  $x \in \mathbb{R}^n$ ,  $x = \{x_1, x_2, \dots, x_n\}$ , neste trabalho, as *features* utilizadas são indicadores econômicos, monetários, sociais e da construção civil descritos na seção ??, já o processamento realizado pelos modelos de regressão pode ser descrito pela função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ .

Os modelos de classificação, por outro lado, têm como objetivo determinar a qual das  $k$  categorias disponíveis um *input* pertence de acordo com **dl-oreilly**. Dessa forma, é utilizada uma função  $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$ , quando  $f(x) = y$ , o vetor de entrada  $x$  foi classificado na categoria  $y$ . Um exemplo de tarefa de classificação, seria determinar se uma operação com cartão de crédito é fraudulenta ou não, neste trabalho, contudo, não foi empregado tal tipo de modelo.

Ainda segundo **Goodfellow-et-al-2016**, os problemas de aprendizado de máquina também podem ser divididos entre aprendizado não-supervisionado e supervisionado. No primeiro, o modelo recebe um conjunto de dados (*dataset*) não rotulado e aprende propriedades da estrutura dos dados, um exemplo de aprendizado não supervisionado é a clusterização, que consiste em dividir o conjunto de dados em *clusters* com amostras similares. No último, por sua vez, os dados de entrada estão associados a resultados conhecidos, chamados de *labels* ou rótulos. Neste trabalho, utiliza-se aprendizado supervisionado para

prever o consumo de cimento mensal nos estados da União a partir dos dados de entrada e compará-lo com o valor real do consumo e, assim, calcular a precisão do modelo.

## 1.3 Modelos utilizados

Neste trabalho, utilizou-se três categorias de modelos de aprendizado de máquina para prever a demanda por cimento: regressão linear, redes neurais *multi layer perceptron* e redes recorrentes. Foram testados, também, três métodos de pré-processamento de dados (*standard scaler*, *minmax scaler* e *power transformer*) e diferentes arquiteturas de redes neurais ao alterar o número de camadas, a quantidade de neurônios em cada camada, o tipo de camada, a função de ativação entre outras configurações, com o objetivo de comparar o desempenho dos modelos e encontrar o que apresenta menor erro na previsão.

### 1.3.1 Regressão linear

A regressão linear é um modelo de aprendizado de máquina que assume um relacionamento linear entre a variável que será prevista (*target*) e os dados de entrada. Desse modo, seu objetivo é construir uma função que, para cada par<sup>3</sup>  $(x, y)$ , recebe como entrada o vetor  $x$  que corresponde às variáveis de *input*,  $x \in \mathbb{R}^k$ ,  $x = \{x_1, x_2, \dots, x_k\}$  e calcula coeficientes  $\beta = \{\beta_0, \beta_1, \dots, \beta_k\}$  para cada atributo de  $x$ , além da constante  $\beta_0$ . O algoritmo, então, utiliza esses coeficientes para prever um valor para a variável *target*,  $y$ , sendo que  $y \in \mathbb{R}$ , segundo **forecasting**. Seja, então,  $\hat{y}$  o valor previsto pelo modelo para um par  $(x, y)$ , a função que descreve a regressão linear, então é dada por

$$\hat{y} = f(x) = \beta_0 + \beta_1 x_{1_i} + \beta_2 x_{2_i} + \dots + \beta_k x_{k_i} \quad (1.1)$$

É possível, logo, contruir uma matriz  $X$  em que a linha  $i$  corresponde ao vetor  $x_i$  dos dados de entrada e cada coluna  $j$  representa uma *feature*. Pode-se, além disso, construir a matriz  $\beta$  dos coeficientes associados a cada elemento da matriz  $X$ . Assim, o modelo é dado por:

$$y = X\beta + \epsilon \quad (1.2)$$

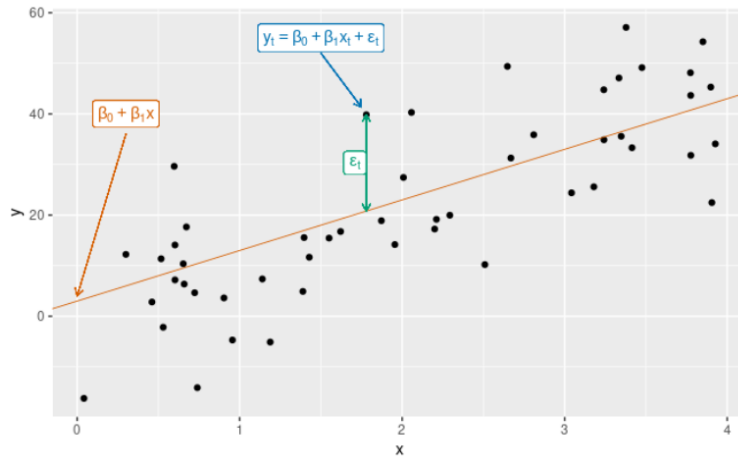
Na equação ??,  $\epsilon$  é o vetor com o erro associado a cada umas das previsões, tal que  $\epsilon_i = y_i - \hat{y}_i = y - (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)$ . Os coeficientes  $\beta$  são calculados durante o treinamento do modelo utilizando o método do *least squares estimation* que visa minimizar a soma do erro quadrado associado às previsões, como descrito em:

$$\sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i) = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{1_i} + \beta_2 x_{2_i} + \dots + \beta_k x_{k_i})) \quad (1.3)$$

<sup>3</sup> No caso deste estudo, o par  $(x, y)$  é tal que  $x$  representa o valor de cada indicador descrito na seção ?? em um estado e mês, já  $y$  corresponde ao número de toneladas de cimento consumidas por esse estado no mês e ano correspondente.

Os coeficientes  $\beta_1, \beta_2, \dots, \beta_k$  determinam como cada atributo de entrada afeta a previsão. Por exemplo, se o coeficiente da variável  $x_i$  for  $\beta_i = 0$ , essa variável não tem influência no valor previsto pelo modelo. Caso o coeficiente seja positivo, por outro lado, um aumento no valor de  $x_i$  resulta em aumento no valor previsto  $\hat{y}_i$ , já se  $\beta_i$  for negativo um aumento no valor de  $x_i$  se reflete na diminuição no valor de  $\hat{y}_i$ .

Na imagem ??, há um exemplo de um modelo de regressão linear com apenas uma variável:



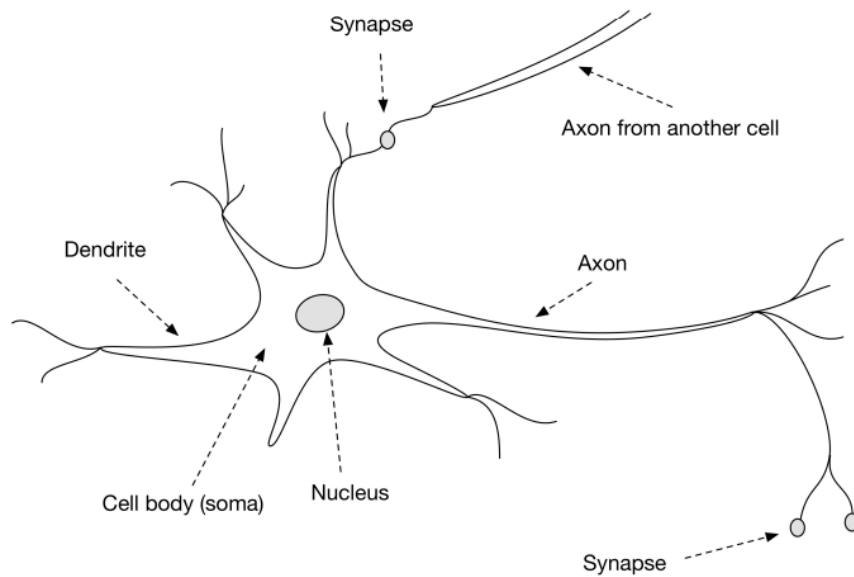
**Figura 1.2:** Exemplo de um modelo simples de regressão linear (forecasting)

Na figura ??, as observações  $y_i$ , estão representadas pelos pontos pretos, enquanto a linha em laranja corresponde à previsão realizada pelo modelo. Observa-se que o modelo não prevê com total exatidão os dados observados, há um erro associado a cada previsão, como o destacado em verde na ilustração, comportamento esperado já os fenômenos previstos são sujeitos a fatores externos e não lineares como os modelos matemáticos.

Por se tratar de um modelo mais simples, os resultados obtidos com a regressão linear são utilizados neste trabalho como base para comparar o desempenho de modelos mais robustos.

### 1.3.2 Redes neurais

Redes neurais são modelos computacionais inspirados no funcionamento do cérebro animal. O cérebro é formado por neurônios que se conectam para transmitir informações sem a necessidade de uma unidade central de controle. Um neurônio biológico, então, é uma célula nervosa que se comunica com outros neurônios por meio de impulsos eletroquímicos. Essa comunicação é denominada sinapse e ocorre apenas se o impulso for forte o bastante para ativar a liberação de químicos na fenda sináptica, de acordo com segundo **deeplearningbook**. Um neurônio é composto de vários dendritos, um axônio e um corpo celular, como ilustrado em ??.



**Figura 1.3:** Ilustração de um neurônio biológico (dl-oreilly)

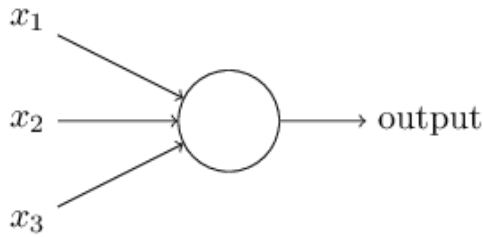
Destaca-se na figura ??, informação chegando a um dendrito do neurônio em destaque por meio de uma sinapse, além da comunicação com outra célula por meio de outra sinapse iniciada no axônio do neurônio em questão.

O processo de propagação da informação nos neurônios envolve as três partes da célula: dendritos, corpo celular e axônio. Os dendritos recebem informações de neurônios vizinhos na forma de impulsos elétricos e são responsáveis por conduzi-las até o corpo celular. Ao chegar no local, a informação é processada e novos impulsos são gerados e repassados a outro neurônio através do axônio no processo de sinapse, segundo **fund\_deep\_learning**. A estrutura e funcionamento dos neurônios biológicos inspiraram os cientistas ao projetarem neurônios artificiais, como os *perceptrons*.

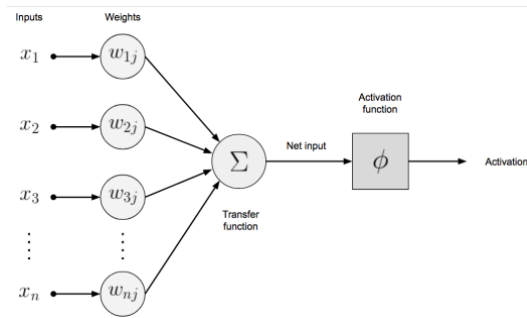
### Neurônios artificiais

O *perceptron*, foi desenvolvido em 1957 por Frank Rosenblatt, inspirado nos trabalhos de Warren McCulloch e Walter Pitts <sup>4</sup>. Trata-se de um modelo linear de classificação que recebe  $n$  entradas e produz uma saída binária, como mostrado na ilustração simplificada ??, conforme **deeplearningbook**. Esse modelo inicial apresentava limitações e foi evoluído com o passar do tempo, contudo, as redes neurais atualmente utilizam, em geral, outro modelo de neurônio como ilustrado em ??.

<sup>4</sup> Em 1943, Warren McCulloch e Walter Pitts, em **neuronio**, apresentaram a primeira ideia de neurônio artificial.



(a) Visão simplificada de um neurônio (deeplearningbook)



(b) Ilustração da arquitetura de um neurônio artificial (dl-oreilly)

A figura ?? mostra o funcionamento de um neurônio artificial. O neurônio apresenta  $n$  entradas  $x_i$ , cada uma associada a um peso  $w_i$ , que expressa a importância das respectivas entradas para o valor de saída<sup>5</sup>. O produto escalar entre os pesos e as respectivas entradas, chamada de *net input* na imagem ??, passa por uma função de ativação<sup>6</sup>  $\phi$  que determina a saída do neurônio. Além disso, um valor de *bias* ou polarização é adicionado ao produto escalar e possibilita que um neurônio que possua todas as entradas nulas apresente saída não nula, dessa forma, aumenta a capacidade de aproximação da rede. **deeplearningbook**

É possível, então, traçar um paralelo entre o neurônio artificial e o biológico. Em primeiro lugar, as entradas  $x_i$  do neurônio têm um funcionamento similar aos dendritos responsáveis por receber as informações que chegam à célula. Em segundo lugar, o processamento matemático entre as entradas e os pesos tem comportamento similar ao corpo celular do neurônio biológico, que processa a informação recebida. Por fim, a função de ativação é responsável por modelar o *output* que será repassado às camadas seguintes como uma das entradas assim como o neurônio repassa, por meio de sinapses, os impulsos processados a outros neurônios.

Seja  $x$  o vetor das  $n$  entradas do neurônio, então  $x \in \mathbb{R}^n$ ,  $x = \{x_1, x_2, \dots, x_n\}$  e seja  $w$  o vetor com os pesos associados a cada entrada,  $w \in \mathbb{R}^n$ ,  $w = \{w_1, w_2, \dots, w_n\}$ . Além disso, seja  $b$  o valor de *bias* e  $\Phi$  a função de ativação. Dessa forma, o *output* de um neurônio é dado por:

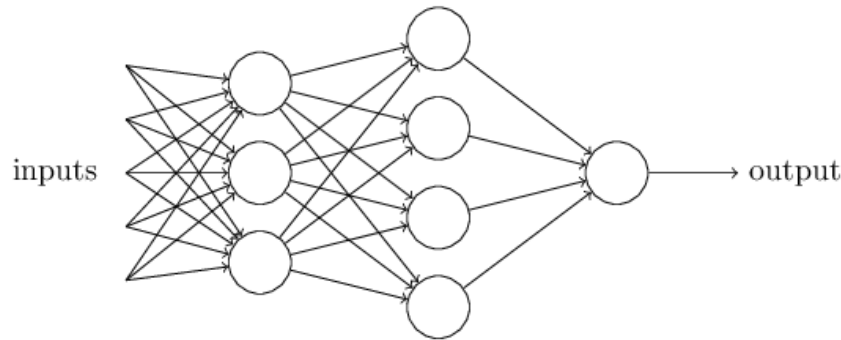
$$h_{w,b} = \Phi(w \cdot x + b) \quad (1.4)$$

Essa saída é utilizada como uma das entradas dos neurônios na camada seguinte, de modo a formar a estrutura das redes neurais.

Os neurônios, então, formam a unidade que compõe as redes neurais artificiais, como ilustrado na figura ??:

<sup>5</sup> O peso atribuído a uma das entradas expressa a influência dessa no *output* do nó, de modo similar aos coeficientes calculados para cada atributo da regressão linear, explicada na seção ??

<sup>6</sup> Um detalhamento sobre as funções de ativação pode ser encontrado na seção ??



**Figura 1.5:** Ilustração de uma rede neural simples.([deeplearningbook](#))

A imagem ?? representa uma simplificação de uma rede neural. Pode-se observar o fluxo da informação, primeiro recebida pelos neurônios de uma camada, então processada nessas unidades e, por fim, repassada à camada seguinte, até obter a saída final da rede. Em particular, é representada uma rede *feed forward*, uma vez que não há conexões entre neurônios de uma mesma camada nem conexões entre uma camada e a anterior. A informação então se propaga apenas no sentido da camada de entrada em direção à final, de saída. As camadas intermediárias são chamadas de camadas ocultas ou *hidden layers*.

### Função de ativação

A função de ativação determina se um neurônio será ativado, ou seja, se a saída propagada para a camada seguinte. Enquanto os pesos e o *bias* realizam uma transformação linear nos dados de entrada, a função de ativação aplica uma transformação não linear, assim, possibilita que a rede neural resolva problemas não lineares e complexos, como reconhecer padrões de escrita, conforme [deeplearningbook](#) e [zhang2021dive](#).

A função de ativação é um atributo de cada uma das camadas da rede e é escolhida de acordo com a tarefa que será executada. Por exemplo, a função sigmóide é recomendada para problemas de classificação. Neste trabalho, foram utilizadas: *rectified linear unit* (ReLU) e *swish*.

### Rectified linear unit (ReLU)

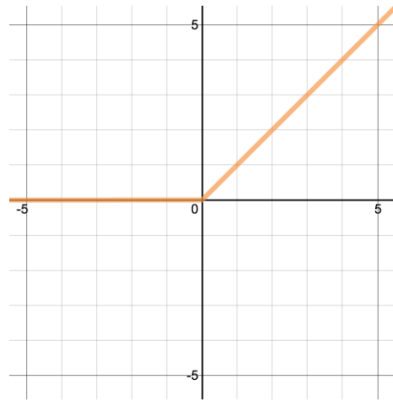
A função ReLU, do inglês *rectified linear unit*, é a função de ativação mais popular atualmente, uma vez que apresenta bom desempenho em diferentes tarefas, de acordo com [dl-oreilly](#). A função ReLU é dada por:

$$f(x) = \max(0, x) \quad (1.5)$$

A derivada da função, utilizada nos modelos de *machine learning* para atualizar os pesos e *bias* no treinamento da rede, é zero, quando a entrada é nula, ou uma constante,



caso contrário. Dessa forma, a ReLU não sofre do problema da dissipação do gradiente<sup>7</sup> como outras funções de ativação. O gráfico da ReLU está ilustrado em ??.



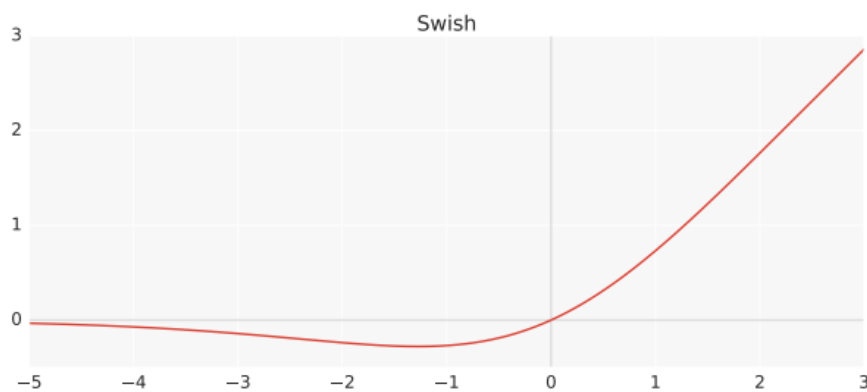
**Figura 1.6:** Rectified linear unit (ReLU) (dl-oreilly)

### Swish

A função *swish* foi proposta por pesquisadores da Google em **swish**, com a promessa de apresentar desempenho igual ou superior à ReLU em redes neurais profundas. A *swish* é uma função não monótona<sup>8</sup> e suave<sup>9</sup>, cuja fórmula é dada por:

$$f(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1 + e^{-x}} \quad (1.6)$$

O gráfico da função é similar ao da ReLU e pode ser observado em ??.



**Figura 1.7:** Função swish (swish)

<sup>7</sup> Um detalhamento sobre o problema da dissipação do gradiente está na seção ??.

<sup>8</sup> Uma função definida como  $f : I \rightarrow \mathbb{R}$  é monótona se é não crescente no intervalo  $I$  ou não decrescente em  $I$ .

<sup>9</sup> Uma função  $f$  é suave se possui derivada de todas as ordens.

### Problema da Dissipação do Gradiente

De acordo com **neuralnetworksanddeeplearning** aprendizado de uma rede neural se dá ao alterar os valores dos pesos e *bias* dos neurônios de acordo com a função de custo, também conhecida como *loss function*. Sejam  $x$ ,  $w$  e  $b$  os conjuntos de entradas, pesos e *bias* da rede, respectivamente, a função de custo, então, é dada por:

$$C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2 \quad (1.7)$$

Na equação da função de custo ??,  $y(x)$  representa a resposta que a rede tenta prever,  $a$  representa o vetor de *output* da rede e  $n$  é o número total de entradas de treino. É possível, então, calcular o gradiente associado ao  $j$ -ésimo neurônio da  $i$ -ésima:

$$\delta_j^i = \frac{\partial C}{\partial b_j^i} \quad (1.8)$$

Os gradientes dos neurônios de uma camada determinam quão rápido essa camada aprende, ou seja, determinam a variação dos valores dos pesos e *bias*. Dessa forma,  $\partial^i$  é o vetor cujos elementos determinam a velocidade do aprendizado da camada  $i$ , então  $\|\partial^i\|$  representa a velocidade de aprendizado da camada  $i$ .

O problema da dissipação do gradiente<sup>10</sup> ocorre, então, quando os gradientes das camadas iniciais, durante o treinamento da rede, ficam com valores muito próximos a zero, dessa forma, o aprendizado das camadas iniciais torna-se muito lento e custoso para a rede.

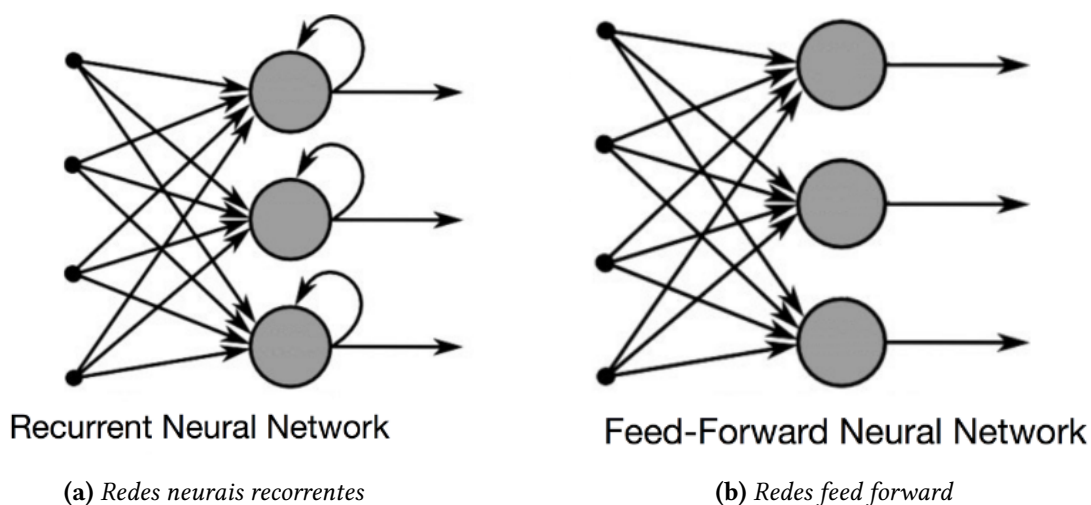
### 1.3.3 Redes neurais multi-layer perceptrons

Redes neurais são modelos de *machine learning* inspiradas no cérebro humano aonde o aprendizado ocorre ao se agregarem neurônios matemáticos que estabelecem conexões de acordo com o treinamento fornecido. Neste trabalho, aplicaram-se redes *multilayer perceptrons* (MLPs), ou seja, que apresentam múltiplas camadas de neurônios e *feedforward*, onde a saída de uma camada de neurônios é utilizada como entrada para a camada seguinte, sem utilizar retropropagação.

### 1.3.4 Redes Neurais Recorrentes

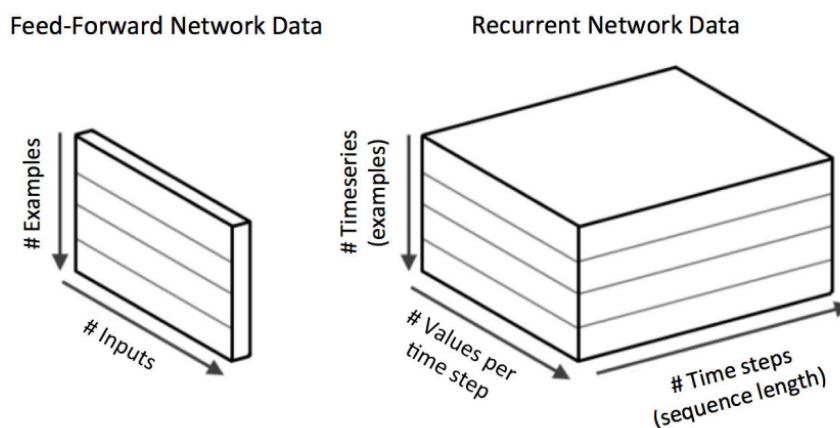
Se por um lado as redes neurais tradicionais são projetadas para realizar previsões sem modelar dimensão temporal, as redes neurais recorrentes se diferem por capturar informação de contexto, ou seja, histórico, para realizar previsões. Essa dimensão temporal, então, se dá por meio de ciclos nas conexões, ou *feedback loops*, como mostrado na figura abaixo.

<sup>10</sup> O problema da dissipação do gradiente é conhecido em inglês como *The vanishing gradient problem*.



Dessa forma, os neurônios nas redes recorrentes recebem entradas de ativação de seu próprio estado atual e dos nós anteriores na rede, não apenas dos anteriores na rede, como nas redes tradicionais.

As redes recorrentes, então, não recebem como entrada apenas um vetor de tamanho fixo como as redes *feed forward*, seu *input* é formado por vários vetores, um para cada *time-step*. Na figura abaixo, o número total de amostras corresponde ao campo "examples", o vetor com as variáveis de entrada ao campo "inputs" ou "values per time step" (na figura de Recurrent Network Data) e o tamanho das sequências consideradas pelo modelo ao "time steps".



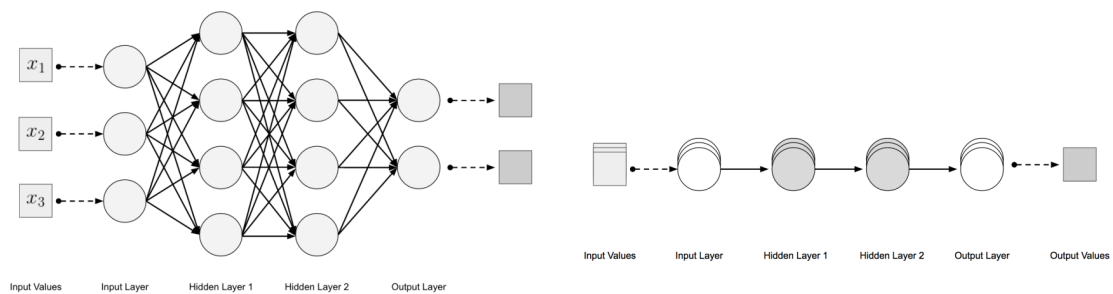
**Figura 1.9:** Comparação entre a entrada de uma rede neural tradicional e um recorrente **dl-oreilly**

Redes recorrentes são uma subclasse das redes neurais, contudo, possuem mais de uma categoria, neste trabalho testaram-se as redes LSTM e GRU.

## LSTM

As redes Long Short Term Memory (LSTM) foram introduzidas em 1997 por Hochreiter and Schmidhuber<sup>11</sup> e são a variação mais utilizada das redes neurais recorrentes atualmente, por exemplo para classificar séries temporais, reconhecimento de fala e de escrita a mão. Redes LSTM apresentam células de memória conectadas, o conteúdo dessas células é modulado pelo *input gate* e o *forget gate*, por exemplo, se ambos estiverem fechados em um tempo o conteúdo da célula permanecerá o mesmo no tempo em questão e no próximo. Essa estrutura permite que a informação seja retida ao longo do tempo e evita o problema do *vanishing gradient* que ocorre com a maioria das redes recorrentes.

As redes neurais LSTM apresentam uma arquitetura diferente das redes tradicionais, uma vez que há ciclos de *feedback* nas conexões entre as células. Para melhor ilustrar essa arquitetura, **dl-oreilly** utiliza a visualização *flat* ou achatada das redes neurais, na qual as células de uma mesma camada da rede estão representadas como um único nó. Pode-se observar essa representação em ?? está a representação tradicional de uma rede neural *feed forward*, já em em ?? está a mesma rede com a representação achatada.

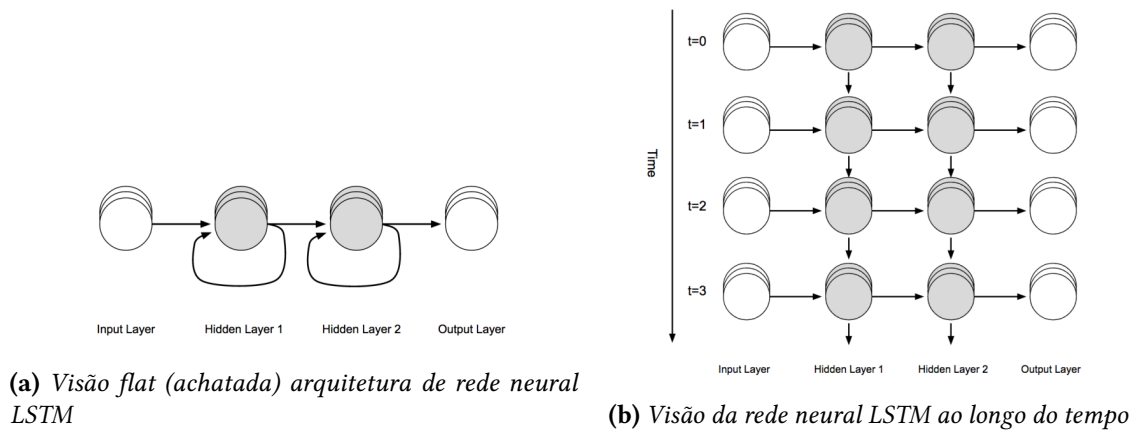


(a) Arquitetura de rede neural feed forward

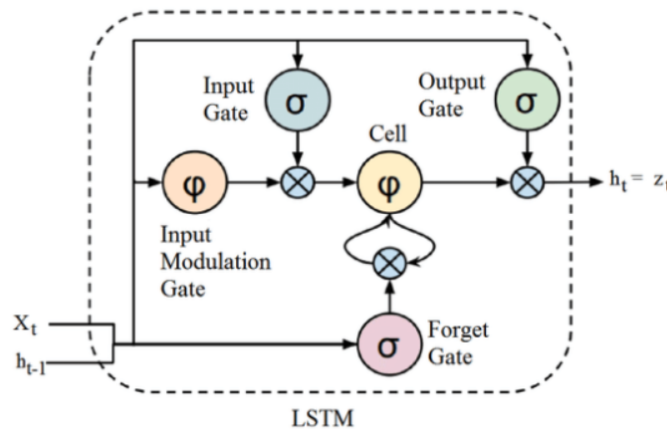
(b) Visão flat (achatada) arquitetura de rede neural feed forward

As redes LSTM introduzem o conceito de conexão entre a saída de uma das camadas ocultas da rede neural com a entrada da mesma camada. A partir desse ciclo, obtém-se entradas de tempos anteriores como parte da informação que chega à rede neural no tempo atual. Na figura ??, essas conexões recorrentes são representadas como as setas que saem de uma célula e atingem a mesma célula, uma vez que utiliza-se a representação achatada. A imagem ??, por sua vez, representa a rede LSTM desenrolada através do eixo do tempo.

<sup>11</sup> lstm-origem



As unidades que formam cada uma das camadas de uma LSTM são uma variação dos neurônios artificiais clássicos. Essas unidades, representadas em ??, permitem que a rede mantenha o estado ao longo do tempo e apresentam dois tipos de conexões: conexões vindas da camada anteriores e vindas *outputs* dessa mesma unidade em tempos anteriores.



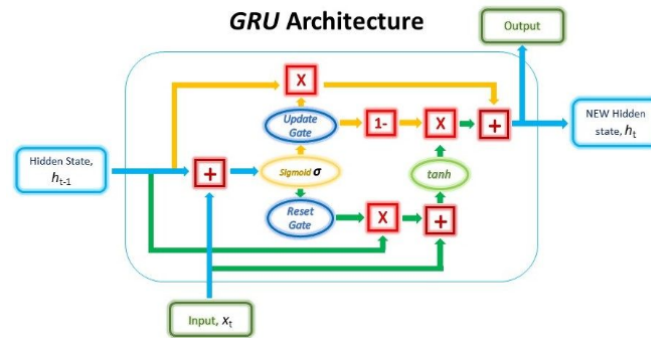
**Figura 1.12: Célula de memória de rede LSTM deeplearningbook**

A unidade LSTM recebe duas entradas, a observação no tempo atual e a saída do último estado oculto. Nas unidades LSTM, a informação é retida nas células e as manipulações de memória são realizadas nos *gates* (portões). O *forget gate* é responsável por remover informações que não são mais úteis para a célula, ou seja, pelo esquecimento, essa operação é realizada por meio de multiplicação de matrizes de pesos, adição de parâmetro de viés e função de ativação que fornece uma saída binária. O *input gate*, por sua vez, adiciona informações úteis ao estado da célula a partir da aplicação de funções sigmóides e tangente hiperbólica, além da multiplicação de vetores. Já o *output gate* extrai informações úteis ao estado da célula atual para apresentar como saída da célula e entrada para a próxima com funções tangente hiperbólica e sigmoide, além da multiplicação de vetores.

**deeplearningbook**

## GRU

As GRU, sigla para *gated recurrent units* no inglês, são unidades recorrentes similares às LSTM. Enquanto as unidades das redes LSTM apresentam dois estados passados entre as células, estado da célula e estado oculto, que carregam memória, respectivamente, de longo e curto prazo, as unidades GRU apresentam apenas um estado oculto carregado ao longo do tempo e capaz de manter as dependências de curto e longo prazo ao mesmo tempo. A estrutura das unidades GRU pode ser vista em ??.



**Figura 1.13:** Unidade da rede GRU deeplearningbook

As unidades GRU apresentam portões, assim como as unidades LSTM, para controlar o fluxo de informações. O *reset gate*, ou portão de redefinição, é responsável por controlar quais informações das etapas anteriores serão mantidas com as da etapa atual por meio da função sigmoide e multiplicação de vetores. O *update gate*, por sua vez, tem como objetivo determinar quanto das informações armazenadas no estado oculto anterior devem ser retidas para o futuro.

### 1.3.5 Normalização

O objetivo da normalização dos dados, é garantir que as variáveis de entrada estejam na mesma escala. Esse processo é necessário não apenas para realizar melhor comparação entre variáveis com diferentes unidades, mas também para o melhor funcionamento dos algoritmos de aprendizado, uma vez que se as *features* estiverem em escalas diferentes os pesos podem ser atualizados mais rápidos que outros.

#### *Standard scaler*

O *standard scaler* garante que as variáveis de entrada estejam em uma escala com as propriedades de uma distribuição normal, ou seja, média ( $\mu$ ) igual a zero e variância ( $\sigma$ ) igual a um. Dessa forma, a operação realizada para aplicar o processo em um entrada  $x$  é

$$z = \frac{x - \mu}{\sigma} \quad (1.9)$$

***Min-Max scaler***

O *min-max scaler* transforma os dados de modo que a escala dos dados tenha um intervalo definido, em geral, entre 0 e 1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1.10)$$





## Capítulo 2

# Metodologia

Neste trabalho, utilizou-se dados públicos de fontes como o Instituto Brasileiro de Geografia e Estatística (IBGE), a Fundação Getúlio Vargas, FGV, e o Banco Central do Brasil (BACEN) para prever o consumo mensal de cimento nos estados da União. Para melhorar o desempenho dos modelos de aprendizado de máquina, realizaram-se técnicas de preparação de dados, como o preenchimento de valores nulos, e para mensurar o acerto das previsões são aplicadas as métricas estatísticas: *root mean squared error* (RMSE), *mean absolute error* (MAE) e *mean absolute percentage error* (MAPE), além da variação percentual.

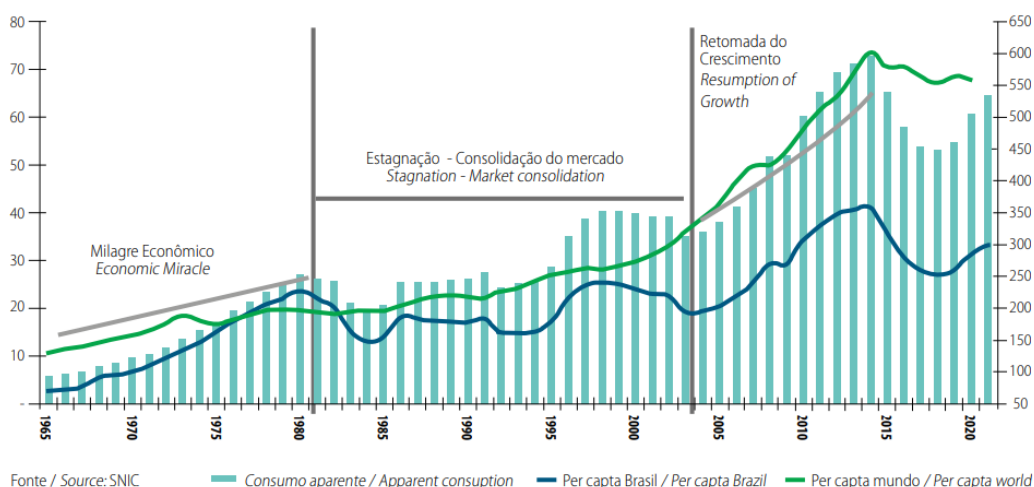
### 2.1 Dados

O consumo mensal de cimento nos estados do Brasil, alvo da previsão dos modelos aplicados neste trabalho, é um indicador fornecido pelo SNIC, Sindicato Nacional da Indústria de Cimento. Os dados obtidos apresentavam granularidade mensal para cada um dos estados da União e estavam disponíveis de 1990 até 2021, a exceção dos anos de 2001 e 2002. Dessa forma, optou-se por utilizar apenas dados posteriores a 2003 no trabalho, uma vez que os modelos recorrentes se valem do contexto para obter a próxima previsão ?? , então esse lapso nos dados poderia trazer inconsistências no modelo.

Além disso, o descarte de dados mais antigos favorece a assertividade do modelo, pois houve alteração no total de cimento consumida no Brasil ao longo do tempo, então dados mais antigos poderiam privilegiar tendências mais antigas. Retirar dados mais antigos, então evita, por exemplo, que a estagnação econômica da época de hiperinflação ou o otimismo do Plano real impacte a previsão atual. Essa alteração na quantidade de cimento consumida pela União pode ser observada na imagem abaixo.

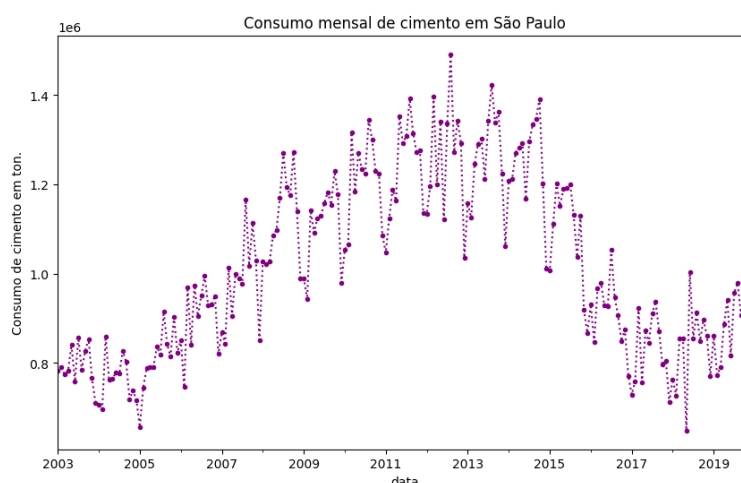
### Consumo aparente no Brasil (em milhões de toneladas)

Apparent consumption in Brazil (millions of tonnes)



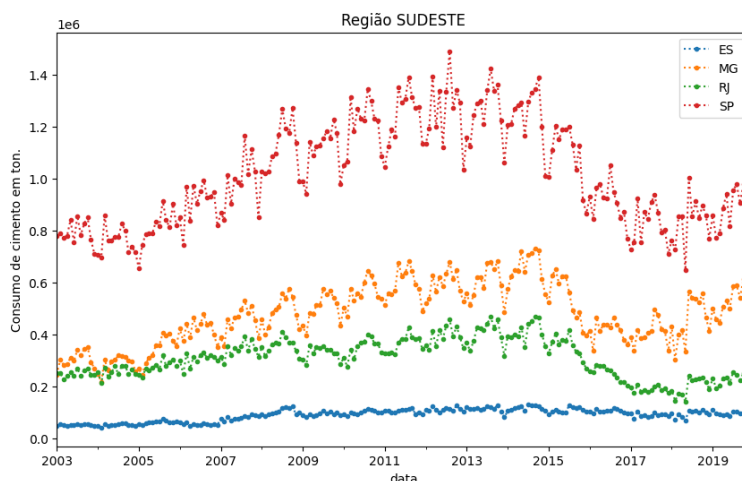
**Figura 2.1:** Evolução do consumo de cimento no Brasil *relatorio-snic*

Observa-se, por exemplo, uma queda no consumo de cimento entre 2015 e 2018, fruto das crises econômicas nesse período. Ressalta-se, também, que o consumo não é uniforme ao longo do ano como pode ser visto na figura ??, aonde observa-se a grande variação na demanda de cimento no estado de São Paulo, não apenas ao longo dos anos, mas também ao longo dos meses do ano



**Figura 2.2:** Evolução do consumo mensal de cimento em São Paulo

Além disso, destaca-se que existe uma grande diferença no consumo entre os estados do Brasil, por exemplo, o consumo médio em São Paulo de 2003 a 2019 foi de aproximadamente 1014,7 mil toneladas enquanto o de Alagoas foi de 41,5 mil toneladas. Mesmo ao comparar apenas estados de uma mesma região, é evidente que há singularidades de cada unidade da federação, como pode ser visto na imagem ??.



**Figura 2.3:** *Evolução do consumo de cimento nos estados do Sudeste*

No gráfico ??, os pontos representam cada uma das medições mensais do consumo de cimento em cada um dos estados da região.

A região Sudeste foi escolhida neste trabalho para demonstrar as previsões do modelo. Essa escolha se deu pois possui apenas quatro estados, então o gráfico com as previsões é mais simples de ler e interpretar que o de outras regiões com maior número de estados. Além disso, os estados dessa região apresentam certa distância na quantidade de cimento consumida, dessa forma, há pouca sobreposição entre as curvas, isso torna o gráfico mais legível também. Por exemplo, a média de consumo de cimento do período analisado no estado de São Paulo foi de 1014,7 mil toneladas, no Rio de Janeiro foi de 310,2 mil toneladas, já em Minas Gerais foi de 480,3 mil toneladas e no Espírito Santo, por sua vez, foi de 91,9 mil toneladas.

O modelo utiliza dados econômicos, sociais e da construção civil para treinar estimar a demanda por cimento. Na tabela ??, estão apresentadas a fonte, granularidade e o período em que os dados estavam disponíveis para todos os indicadores utilizados neste trabalho. Na tabela são utilizadas siglas para melhor visualização, como IBGE para Instituto Brasileiro de Geografia e Estatística, FGV para Fundação Getúlio Vargas, BACEN para Banco Central do Brasil, IPEA para Instituto de Pesquisa Econômica Aplicada e SNIC para Sindicato Nacional da Indústria do Cimento todas fontes utilizadas neste trabalho. Além disso, utilizou-se abreviação para Produto Interno Bruto (PIB), Índice Nacional de Preços ao Consumidor Amplo (IPCA), Índice Nacional de Custo da Construção (INCC), Índice Geral de Preços (IGP), Necessidade de Financiamento do Setor Público (NFSP) e Índice de Desenvolvimento Humano (IDH).

Indicador	Fonte	Período disponível	Granularidade
PIB a preços constantes	IBGE <sup>a</sup>	1983 até 2019	anual por estado
PIB a preços de mercado	IBGE <sup>??</sup>	1985 até 2019	anual por estado
PIB per capita	IBGE <sup>??</sup>	1985 até 2019	anual por estado
PIB da construção civil	IBGE <sup>??</sup>	1985 até 2019	anual por estado
Desemprego	IBGE <sup>??</sup>	1991 até 2022	irregular <sup>b</sup>
IPCA	IBGE <sup>c</sup>	1981 até 2021	mensal para o Brasil
INCC	FGV <sup>d</sup>	1980 até 2021	mensal para o Brasil
IGP	FGV <sup>??</sup>	1944 até 2021	mensal para o Brasil
Taxa Selic	IBGE <sup>e</sup>	1986 até 2022	mensal para o Brasil
NFSP	BACEN <sup>??</sup>	1991 até 2022	mensal para o Brasil
Estoque líquido de capital fixo	IPEA <sup>??</sup>	1947 até 2019	anual para o Brasil
População	IBGE <sup>f</sup>	1991 até 2021	anual por estado
IDH	IBGE <sup>??</sup>	1991 até 2017	irregular <sup>g</sup>
Produção mensal de cimento	SNIC <sup>h</sup>	2003 até 2022	mensal por estado
Valor médio do cimento <sup>i</sup>	IPEA <sup>??</sup>	1947 até 2019	anual para o Brasil

**Tabela 2.1:** Indicadores utilizados no trabalho

<sup>a</sup> Dado retirado do portal do Ipeadata em <http://www.ipeadata.gov.br/Default.aspx>

<sup>b</sup> Havia dados de 1992 até 2014 com granularidade anual e por estado, contudo, a partir de 2012 foram disponibilizados dados mensais a nível de Brasil por conta da Pesquisa Nacional por Amostra de Domicílios (PNAD) Contínua mensal realizada pelo IBGE. Neste trabalho, utilizou-se os dados anuais até 2012 e, após 2012, os dados provenientes da PNAD Contínua

<sup>c</sup> Dado retirado do IBGE em <https://sidra.ibge.gov.br/tabela/1737>

<sup>d</sup> Dado obtido a partir do portal da FGV em <https://www.debit.com.br/tabelas/tabela-completa-pdf.php?indice=incc>

<sup>e</sup> Dado obtido em <https://www.debit.com.br/tabelas/tabela-completa.php?indice=selic>

<sup>f</sup> Dado obtido do portal Base dos Dados em <https://basedosdados.org/dataset/br-ibge-populacao>

<sup>g</sup> Os indicadores de IDH (Renda, Longevidade e Educação) estão disponíveis em anos de censo do IBGE (1990, 2000, 2010), há dados, também, de 2014 a 2017 por conta da PNAD Contínua

<sup>h</sup> Dados retirados do portal <http://www.cbicdados.com.br/menu/materiais-de-construcao/cimento>

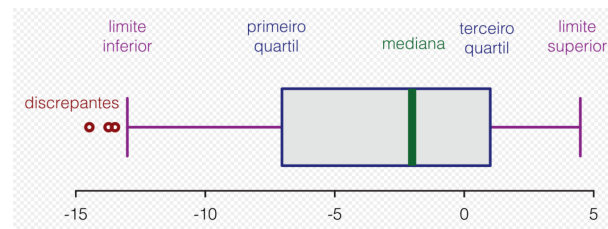
<sup>i</sup> Evolução do valor médio/mediano do cimento Portland 32 em US\$/Tonelada

Com o objetivo de direcionar a estratégia de preparação de dados foi realizada uma análise exploratória de cada um dos dados de entrada e da variável resposta. A partir dessa análise, optou-se por utilizar os dados de 2003 até 2019 para o estudo. Destaca-se nessa análise, a alta taxa de variação dos dados de entrada, em alguns indicadores, por exemplo para o PIB da construção civil, o desvio padrão é maior que o valor médio do indicador. Dessa forma, está presente nos dados um grande número de *outliers*, em especial ao analisar o Brasil como um todo, por conta da forte diferença entre as regiões do Brasil.

*An outlier is an observation strikingly far from some central value. It is an unusual value relative to the bulk of the data...tukey77*

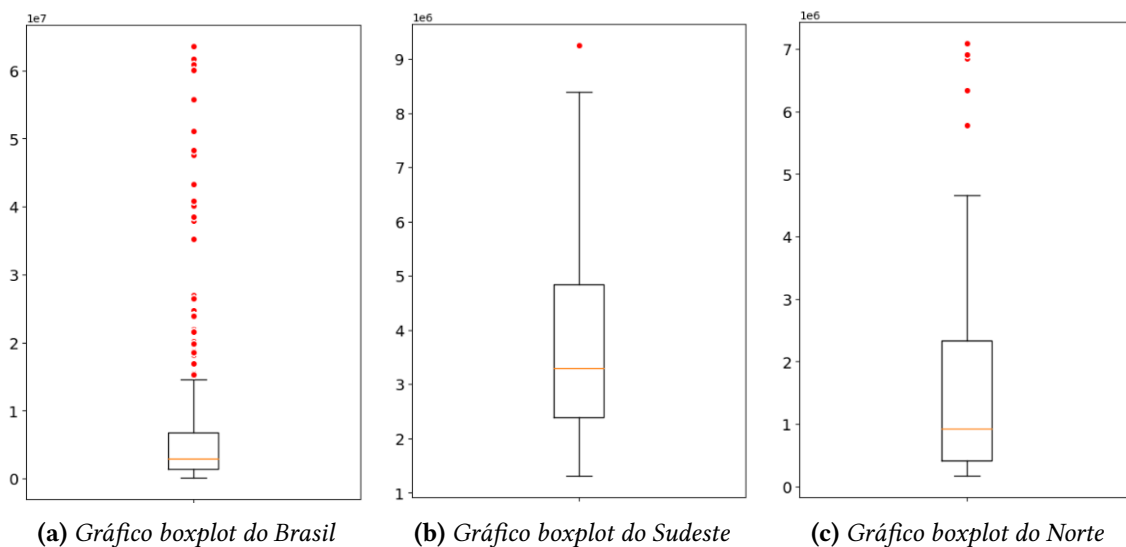
*Outliers* são observações discrepantes do restante dos dados, que parecem inconsistentes e podem interferir no processo de previsão. **outliers** Para ilustrar melhor o alto volume

de *outliers* presentes nos dados, utilizou-se gráficos *boxplot*, como mostrado na figura abaixo.



**Figura 2.4:** Gráfico *boxplot* explicacao-boxplot

O *boxplot* **boxplot** é uma técnica estatística utilizada na análise exploratória os dados para identificar visualmente padrões, como a distribuição dos dados. Na figura, acima linha em verde corresponde à mediana <sup>1</sup> dos dados, os limite inferior e o superior ao retângulo representam o primeiro e terceiro quartil <sup>2</sup>, respectivamente.



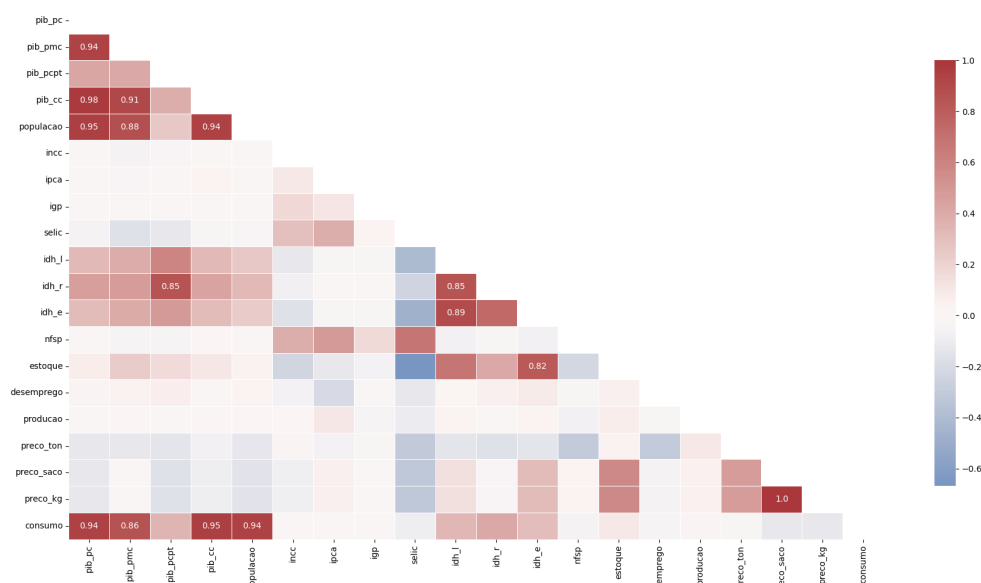
Conforme explicado na figura ??, a linha em laranja nos gráficos acima representa a mediana dos dados, o retângulo que envolve a mediana é delimitado pelo primeiro e terceiro quartil e retém metade central das amostras. Os valores reoesentados por pontos vermelhos acima ou abaixo dos limites superiores ou inferiores são *outliers*. Observa-se, então, que pode-se obter uma significativa redução no número de amostras com *outliers* ao separar a análise por região.**boxplot**

Analisou-se, também, a correlação entre as variáveis de entrada e observou-se alta correlação entre os indicadores de PIB do estado, PIB da construção civil e população.

<sup>1</sup> valor que fica no meio quando os dados estão ordenados ou a média dos dois valores centrais se o número de pontos de dados for par **boxplot-stat**

<sup>2</sup> O primeiro e terceiro quartis também representam a mediana dos valores superiores à mediana dos dados e a mediana dos valores inferiores à mediana. Então metade dos dados está contida dos quadrados nas imagens

Além disso, há alta correlação entre os três indicadores de IDH, além do preço do saco de cimento e o preço do quilograma, como pode-se validar na figura ??.



**Figura 2.6:** Matriz de correlação

Foram adotadas estratégias para garantir dados na granularidade mensal e por estado. Caso os indicadores apresentassem granularidade anual, o valor foi dividido por 12 de modo a obter a média mensal, já caso a granularidade fosse a nível de Brasil, o valor apresentado foi repetido para todos os estados.

A estratégia utilizada para lidar com dados faltantes foi, sempre que possível, repetir o valor anterior que estava disponível nos dados de entrada para preencher a ocorrência. Contudo, alguns indicadores não apresentavam valores mais antigos, então foi usado um valor não presente no intervalo de dados de entrada (-1) para marcar como nulo. Os indicadores com dados faltantes são, em ordem decrescente de acordo com o percentual de dados faltantes: produção de cimento, valor médio do quilo, saco e da tonelada e cimento e desemprego.

Além disso, tomou-se um cuidado para evitar que a previsão fosse realizada com os dados do mês anterior ou do ano anterior no caso dos indicadores anuais. Deslocou-se, portanto, os dados de entrada para frente em um ocorrência de modo a associar os dados de um mês com o consumo no mês seguinte.<sup>3</sup>

Por fim, o estado correspondente à medição foi usado como dado de entrada. Como os modelos de inteligência artificial aceitam apenas caracteres numéricos, utilizou-se o método de codificação *one hot* para criar 27 colunas, uma para cada estado, nas quais o valor é 1 quando a linha possui dados daquele estado e é 0 caso contrário.

<sup>3</sup> os dados correspondentes a, por exemplo, fevereiro de 2004 estão relacionados ao consumo de cimento em março de 2004, com o objetivo de propor um cenário mais pertinente, uma vez que o objetivo do projeto é prever a demanda por cimento no mês seguinte em um estado a partir dos dados do mês atual e, eventualmente, dos anteriores.

## 2.2 Avaliação de performance

Para comparar a eficiência dos modelos mede-se os erros de cada previsão, ou seja, a distância entre o valor previsto pelo algoritmo e o valor do dado real. Neste trabalho, utilizou-se as seguintes métricas estatísticas para mensurar o desempenho: *mean absolute error* (MAE), *root mean square error* (RMSE) e *mean absolute percentage error* (MAPE). Além disso, foi utilizado o delta percentual ( $\Delta$ ) para avaliar se o modelo tende a subestimar ou superestimar o valor previsto, se é otimista ou pessimista.

### 2.2.1 Mean absolute error (MAE)

O MAE, sigla do inglês para *mean absolute error* ou média do erro absoluto mede o erro absoluto de cada previsão e é dado por:

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n} \quad (2.1)$$

### 2.2.2 Root mean squared error (RMSE)

A RMSE, sigla para *root mean squared error* é semelhante à MAE, contudo eleva os erros ao quadrado antes de somá-los e tira a raiz logo depois. A RMSE é, portanto, mais sensível a *outliers*.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (2.2)$$

### 2.2.3 Mean absolute percentage error (MAPE)

Foi utilizada também a MAPE, *Mean absolute percentage error*, para mensurar a escala do erro em relação ao tamanho das medições.

$$MAPE = \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (2.3)$$

### 2.2.4 Delta percentual

O delta percentual,  $\Delta$ , é utilizado para mensurar se o modelo apresenta tendência de subestimar ou superestimar a variável, se é otimista ou pessimista.

$$\Delta = \frac{\hat{y}_i - y_i}{y_i} \quad (2.4)$$





## Capítulo 3

# Experimentos

Para cada um dos modelos selecionados foram realizados experimentos alterando parâmetros e os dados com o objetivo de obter um melhor desempenho. O treino dos modelos foram realizados com dados de 2003 até junho de 2017, os testes, por sua vez, foram realizados com dados de julho de 2017 até dezembro de 2019, dessa forma 85% do conjunto de dados foi usado para treino e 15% para teste. Além disso, cuidou-se para que não houvesse embaralhamento de dados no processo de separação entre treino e teste, uma vez trata-se de dados temporais. Para cada modelo foram alterados diferentes parâmetros.

### 3.1 Regressao linear

A regressão linear é um modelo simples de aprendizado de máquina que visa construir uma função linear entre as variáveis de entrada e o alvo da previsão. Por tratar-se de um modelo mais simples, é utilizado neste estudo como base para comparar o desempenho dos outros modelos. Foram testados métodos de normalização e transformação de dados, além da remoção de variáveis de alta correlação conforme descrito na tabela ??.

Experimento	Transformação nos dados	Período de dados	Remoção de variáveis
A	nenhum	de 2003 até 2019	não
B	<i>Standard scaler</i>	de 2003 até 2019	não
C	<i>MinMax scaler</i>	de 2003 até 2019	não
D	<i>Power Transformer</i>	de 2003 até 2019	não
E	nenhum	de 2003 até 2019	sim <sup>a</sup>

**Tabela 3.1:** Experimentos realizados na regressão linear

<sup>a</sup> As variáveis removidas foram: PIB *per capita*, INCC, IGP, taxa Selic, IDH Educação e IDH Longevidade, NFSP, preço do saco de cimento e preço da tonelada de cimento

O desempenho do modelo em cada um dos experimentos encontra-se na tabela ??:

Experimento	MAE	RMSE	MAPE
A	35148.6	55391.1	0.6
B	35148.6	55391.1	0.6
C	35148.6	55391.1	0.6
D	49073.6	67446.7	1.05
E	<b>26435.6</b>	<b>38970.6</b>	<b>0.32</b>

**Tabela 3.2:** Desempenho dos modelos de regressão linear

Observa-se que não houve alteração na performance ao normalizar os dados com *standard scaler* ou *minmax scaler*, ao utilizar *power transformer*, contudo, houve piora no desempenho. Ressalta-se a grande melhoria nas métricas ao remover variáveis de alta correlação tanto nas métricas de erro quantitativo, como o MAE e RMSE, quanto proporcionalmente ao atingir 32% de erro percentual médio.

Foi realizado o teste de remover variáveis de alta correlação, pois segundo **corr\_reg\_lin**, esse tipo de *feature* pode ter efeitos ruins no modelo como: aumentar a variabilidade dos coeficientes em relação à amostra, ou seja, os coeficientes calculados pelo modelo podem ter um alto grau de variação de uma amostra para outra.

Na sessão ??, estão apresentados os resultados e previsões do modelo de regressão linear com melhor desempenho dentre os testados no modelo, ou seja, o experimento E, sem normalização de dados e removendo *features* com alta correlação.

## 3.2 Redes neurais *Feed Forward*

As redes neurais *feed forward* podem ser construídas com várias arquiteturas e configurações. Neste estudo, testou-se alterar a quantidade de camadas da rede, o número neurônios em cada camada, a função de ativação utilizada, além da quantidade de *epoch*<sup>1</sup> utilizada no treinamento.

## 3.3 Redes recorrentes

## 3.4 regressao linear

<sup>1</sup> Uma *epoch* é uma passada pelos dados de entrada **dl-oreilly**

## Capítulo 4

### Resultados

blafgdf .



## Capítulo 5

## Conclusão

ahahahahahh

