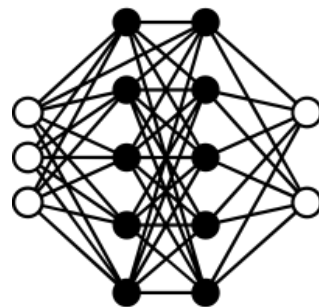


Decisões arquiteturais no contexto de treinamento de modelos de Aprendizado de Máquina

Marianne Monteiro, @hereismari

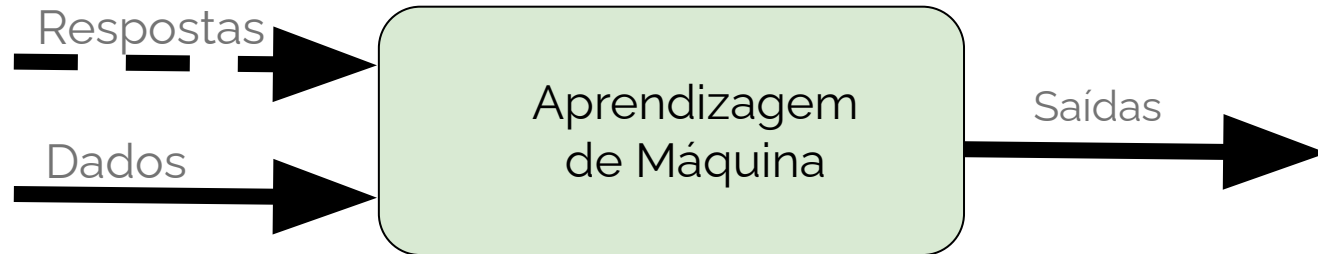


Agenda

- Introdução rápida à Aprendizagem de Máquina (Conceitos básicos, Entidades principais)
- Arquitetura de Software para aplicações de Aprendizado de Máquina
- Escalando o treinamento de modelos (Data parallelism e Model parallelism)
- Conclusões

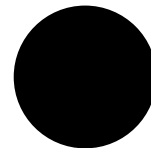
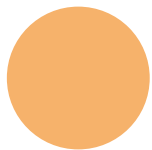
Introdução à Aprendizagem de Máquina

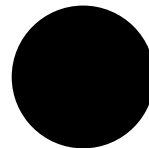
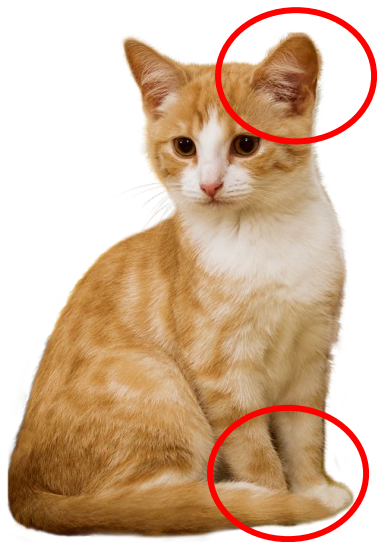
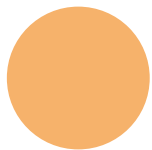
Introdução: aprendizagem de máquina

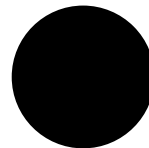
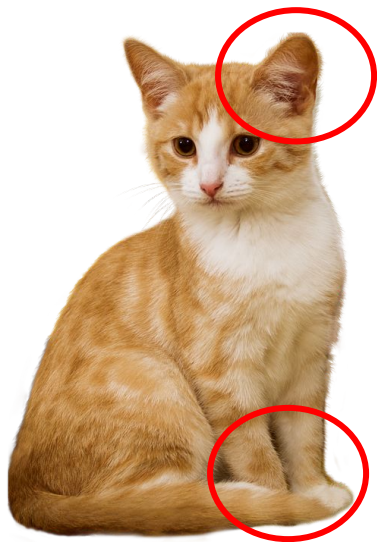
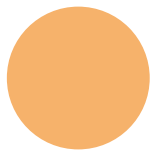














Introdução: aprendizagem de máquina

Um computador aprende a partir da experiência **E** com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E**" - [Tom Mitchell \(1998\)](#)

Introdução: aprendizagem de máquina

Um computador aprende a partir da experiência **E** com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E**" - [Tom Mitchell \(1998\)](#)

Experiência = Dados (Imagens, dados tabulares, áudio, texto, ...)

Introdução: aprendizagem de máquina

Um computador aprende a partir da experiência **E** com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E**" - [Tom Mitchell \(1998\)](#)

Tarefa = Classificação, Regressão, Geração de dados, Tradução, Recomendação, ...

Introdução: aprendizagem de máquina

Um computador aprende a partir da experiência **E** com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E**" - [Tom Mitchell \(1998\)](#)

Performance = erro, acurácia, métricas específicas para tarefa, ...

Introdução: aprendizagem de máquina

Um computador aprende a partir da experiência **E** com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E**" - [Tom Mitchell \(1998\)](#)

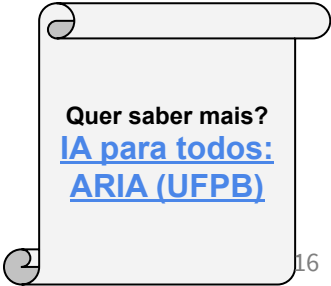
"Um computador" = modelo (um tipo rede neural, árvore de decisão, regressão linear, ...)

Introdução: aprendizagem de máquina

Um computador **aprende** a partir da experiência **E** com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E**" - [Tom Mitchell \(1998\)](#)

Aprende = otimização via Stochastic gradient descent (SGD).

Que é um algoritmo iterativo.

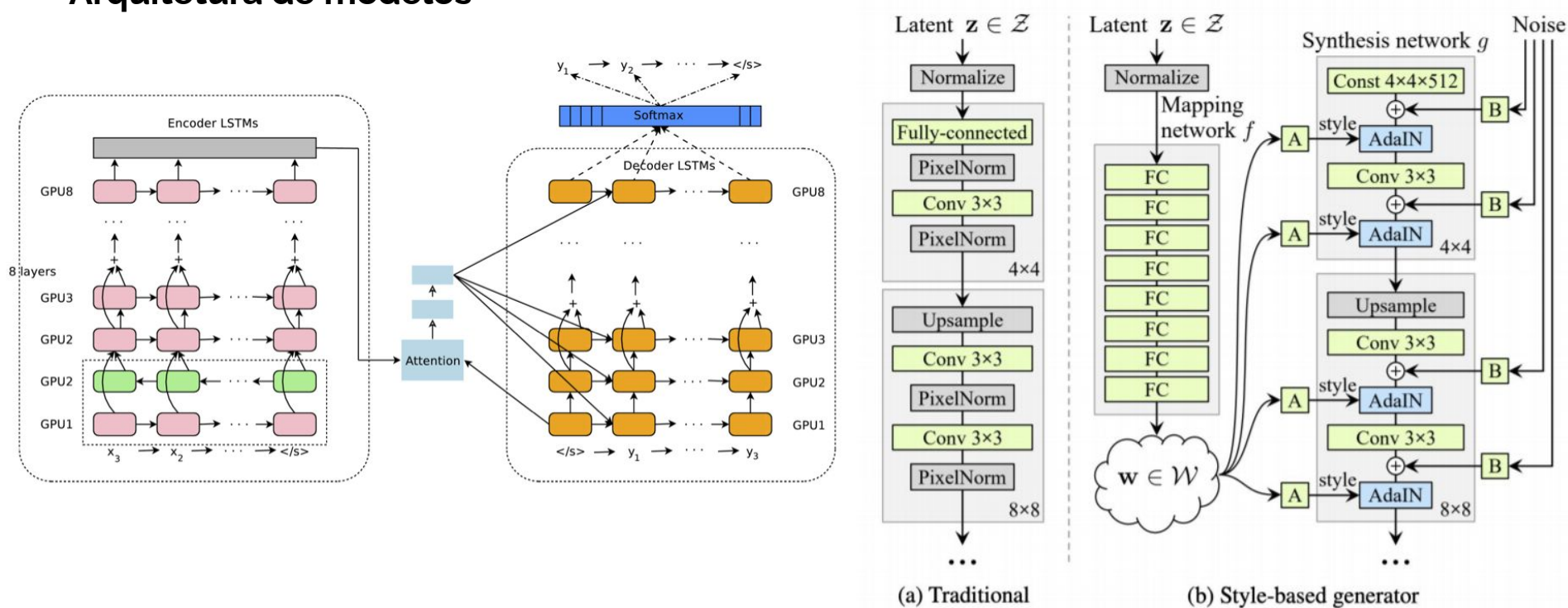


Quer saber mais?
[IA para todos:](#)
[ARIA \(UFPB\)](#)

Introdução: aprendizagem de máquina

Um computador (modelo) aprende a partir da experiência **E** (dados) com respeito a alguma tarefa **T** e alguma medida de performance **P**, se sua performance em **T** melhora com sua experiência **E'** - [Tom Mitchell \(1998\)](#)

Arquitetura de modelos



Imagens de respectivamente x e [\[1812.04948\] A Style-Based Generator Architecture for Generative Adversarial Networks](#)

Arquitetura de Software para aplicações de Aprendizado de Máquina

Exemplo: classificador de texto para análise de sentimentos

"Adorei! Se eu pudesse dar 1000 estrelas não daria nenhuma :)"



Modelo



0.67

Exemplo: classificador de texto para análise de sentimentos

Inferência

"Adorei! Se eu pudesse dar 1000 estrelas não daria nenhuma :)"



Modelo



0.67

Exemplo: classificador de texto para análise de sentimentos

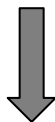
Treino

Textos:

"Excelente serviço, recomendo a todos!"
"Gostei, mas a entrega demorou demais"
"Comida muito boa!"
...

Rótulos:

1
0
1
...



Modelo

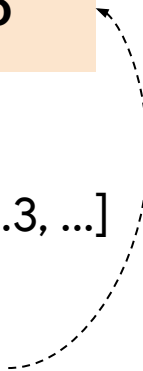


[0.8, 0.4, 0.3, ...]

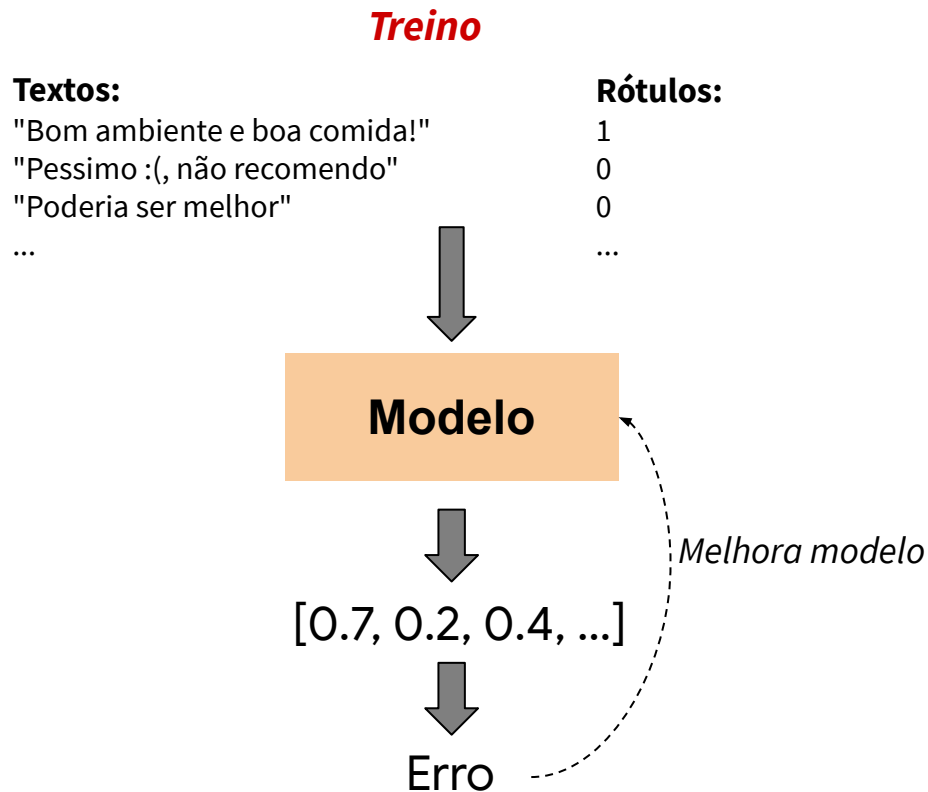


Erro

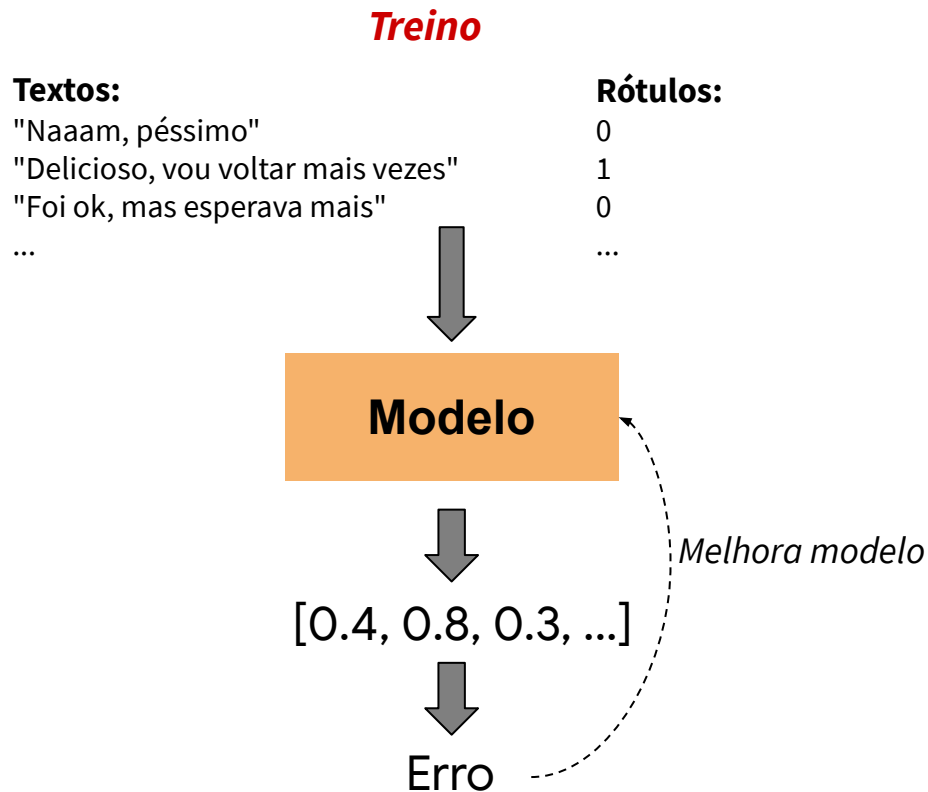
Melhora modelo



Exemplo: classificador de texto para análise de sentimentos



Exemplo: classificador de texto para análise de sentimentos



Exemplo: classificador de texto para análise de sentimentos

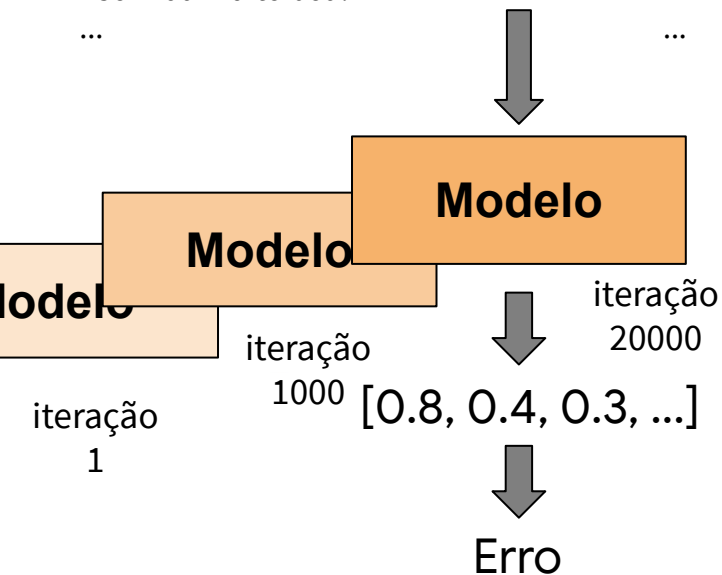
Treino

Textos:

"Excelente serviço, recomendo a todos!"
"Gostei, mas a entrega demorou demais!"
"Comida muito boa!"
...

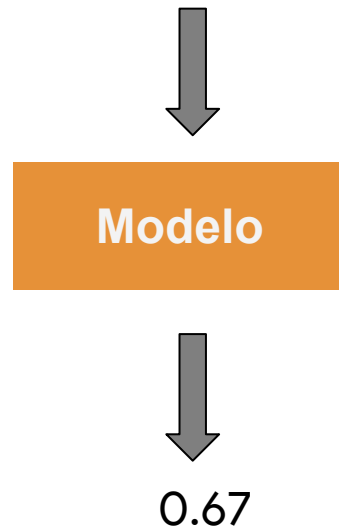
Rótulos:

1
0
1
...

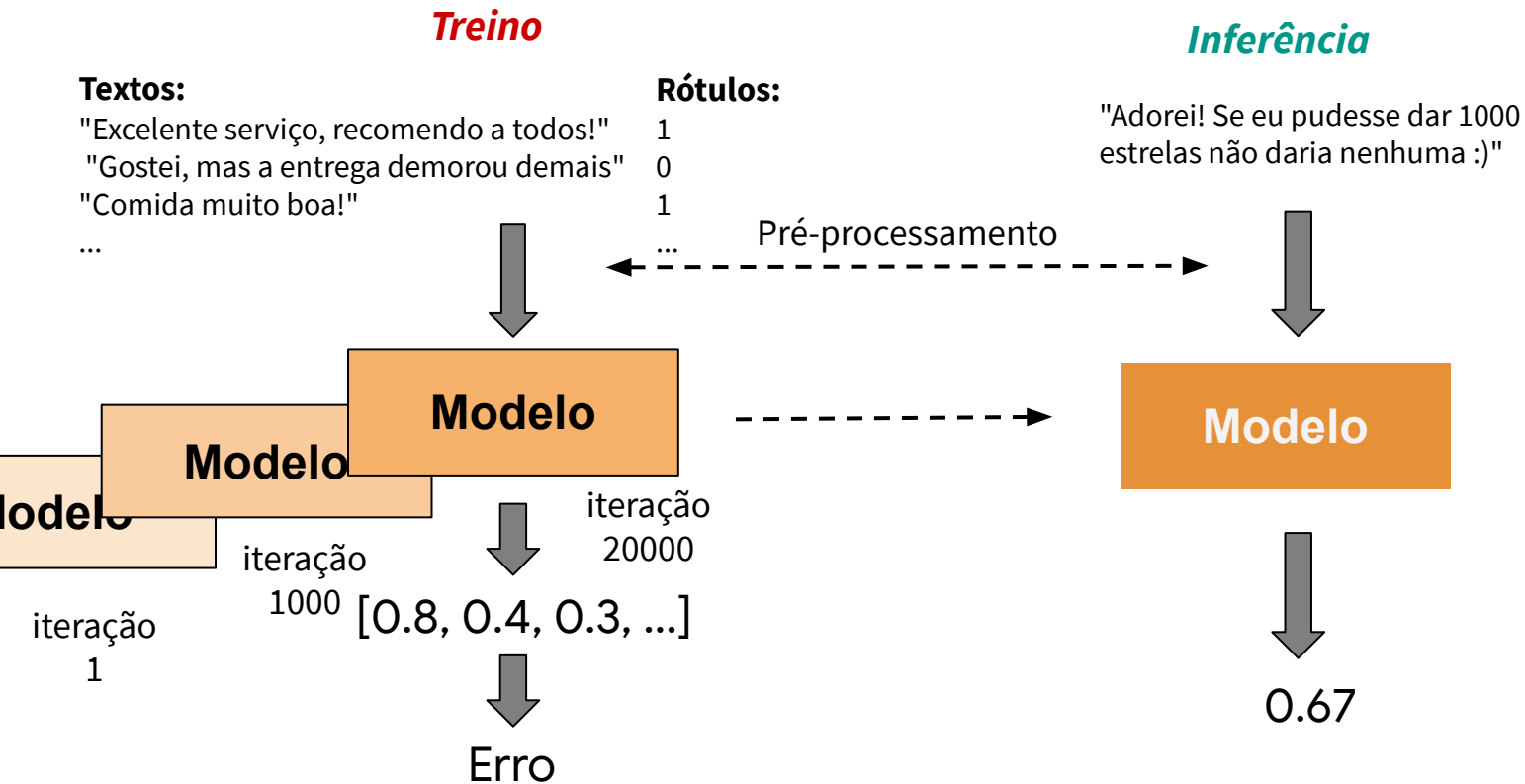


Inferência

"Adorei! Se eu pudesse dar 1000 estrelas não daria nenhuma :)"



Exemplo: classificador de texto para análise de sentimentos



Treino

- Experimental
- O treinamento de um modelo gera vários artefatos que devem ser armazenados:
 - Configuração
 - Diferentes versões do modelo ao longo do treino
 - Métricas ao longo do treino...
- Desenvolvimento rápido, porém experimentação é lenta (o treinamento de um modelo pode levar horas, dias, ...)
- Frameworks de Deep Learning oferecem várias facilidades (ex: tf.data, pytorch, tensorflow)

Inferência (Produção)

- Utilizado em produção
- Métricas devem ser monitoradas
 - Tempo de resposta
 - Modelo está presente? Está sendo usado?
 - Saídas do modelo
 - Acurácia
 - ...
- Requisitos de tempo de resposta, memória, escalabilidade, ...
- Pode utilizar framework ou linguagem de programação diferente do treinamento (ex: Python para treinamento e C++ para servir)

Exemplo: classificador de texto para análise de sentimentos

Inferência

"Adorei! Se eu pudesse dar 1000 estrelas não daria nenhuma :)"

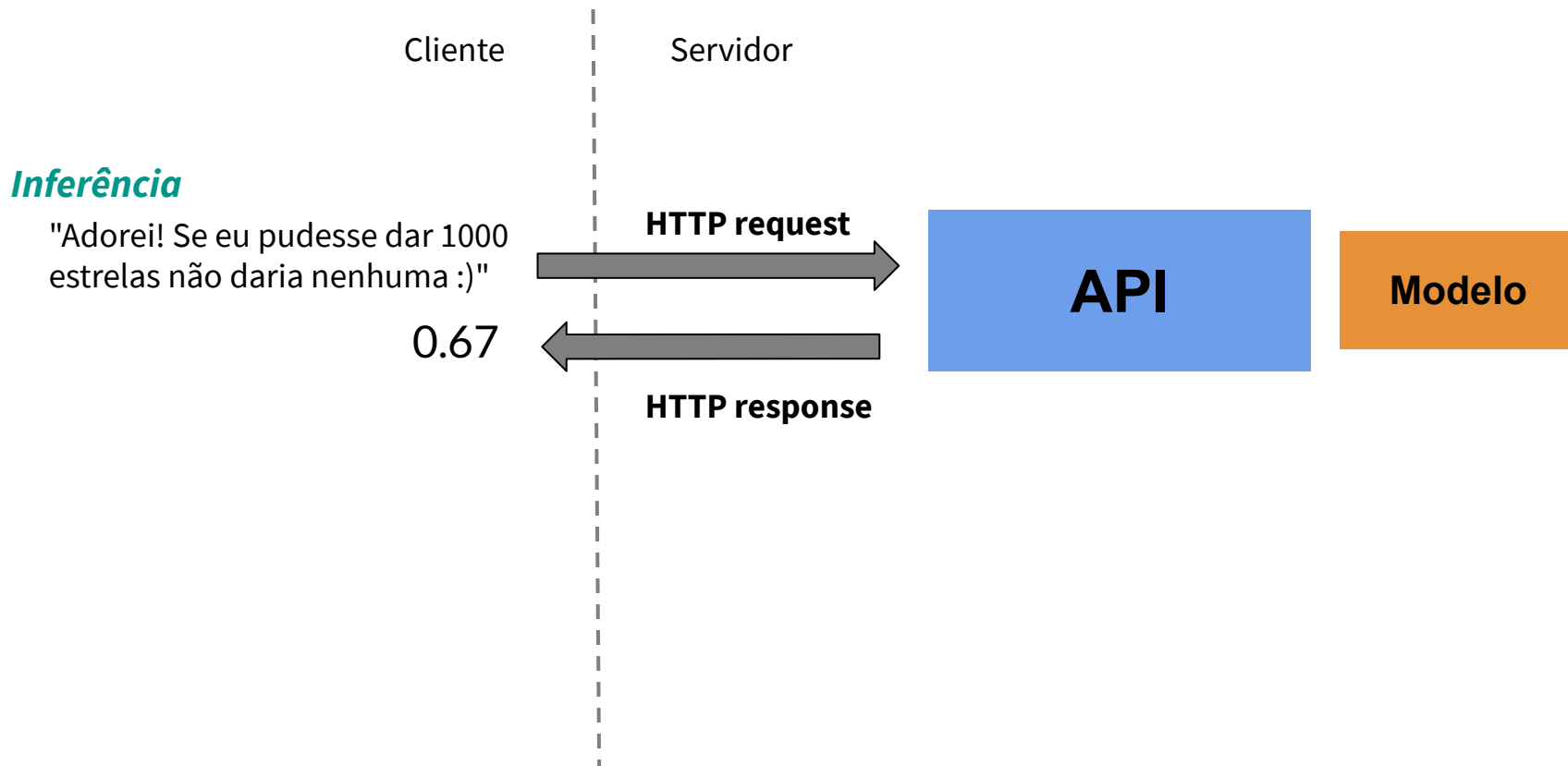


Modelo



0.67

Exemplo: classificador de texto para análise de sentimentos



Exemplo: classificador de texto para análise de sentimentos

Inferência

"Adorei! Se eu pudesse dar 1000 estrelas não daria nenhuma :)"



0.67

Cliente
(dispositivo móvel)

Servidor

Exemplo: classificador de texto para análise de sentimentos

Inferência

"Adorei! Se eu pudesse dar 1000 estrelas não daria nenhuma :)"



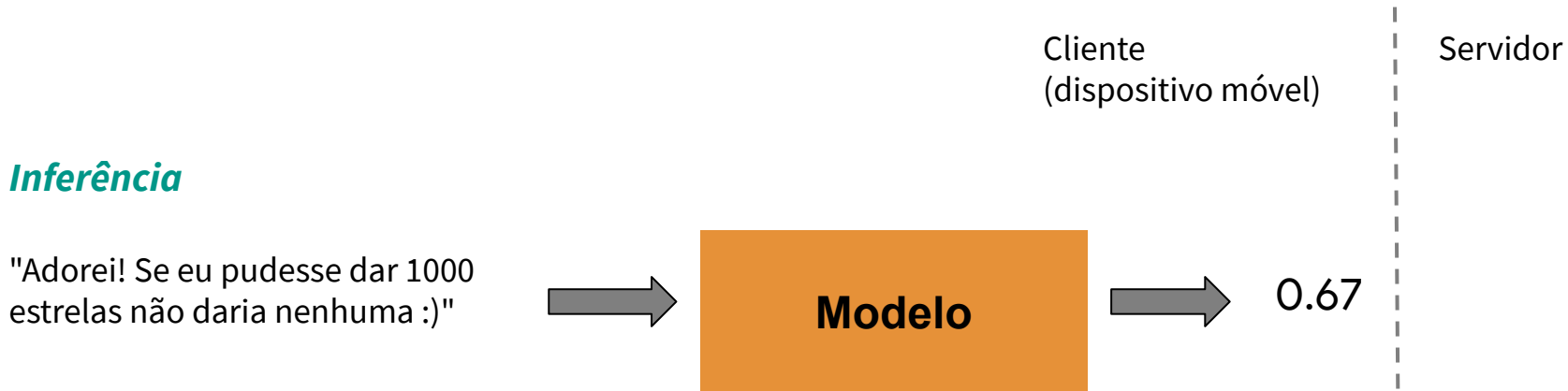
0.67

Cliente
(dispositivo móvel)

Servidor

+ Rápido, privacidade para a cliente!!!

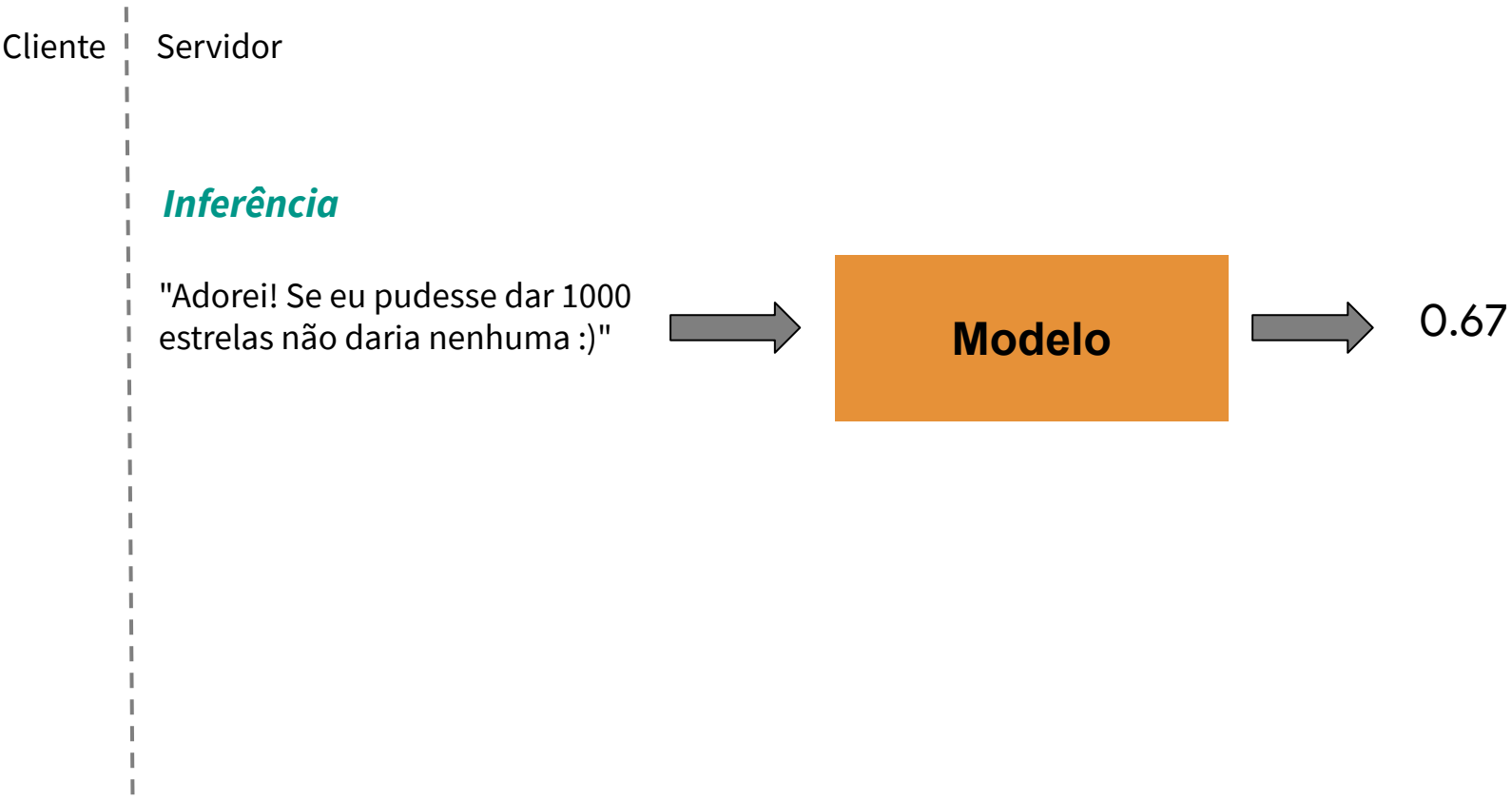
Exemplo: classificador de texto para análise de sentimentos



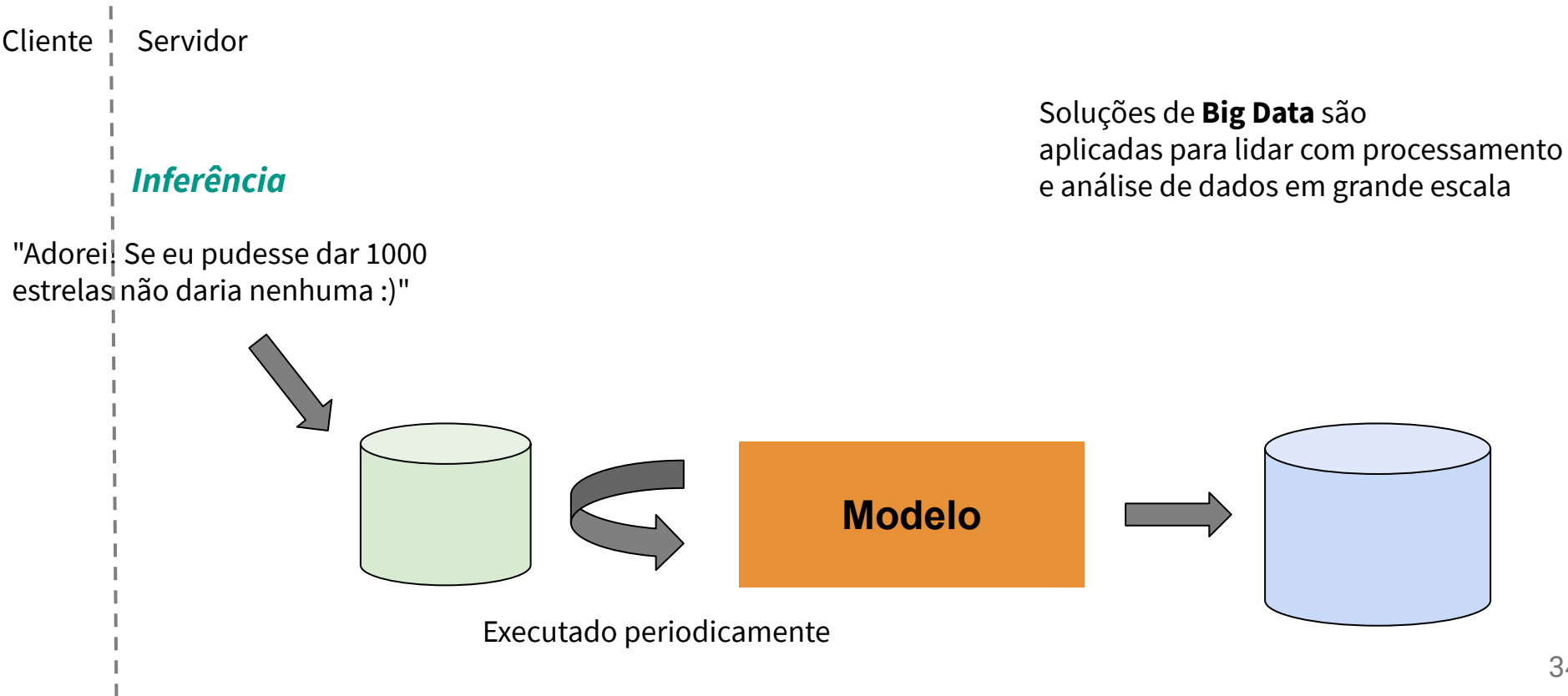
+ Rápido, privacidade para a cliente

- **Complexo, recursos mais limitados, proteger propriedade intelectual**

Exemplo: classificador de texto para análise de sentimentos



Exemplo: classificador de texto para análise de sentimentos



Mas e os dados?

Dados

- Pipeline deve ser bem definida e documentada: de onde vem os dados? Por quanto tempo são armazenados? Como e onde são armazenados? Quem tem acesso? Há versionamento?
- Modificações devem ser amplamente comunicadas.
- **Se houver problema com os dados -> problema em toda pipeline: modelos falham silenciosamente!!**
- Dados devem ser filtrados (análise de qualidade).

Poderia ter uma apresentação só sobre como processar e tomar decisões arquiteturais para lidar com dados...

Escalando o treinamento de um modelo

Escalando o treinamento de um modelo

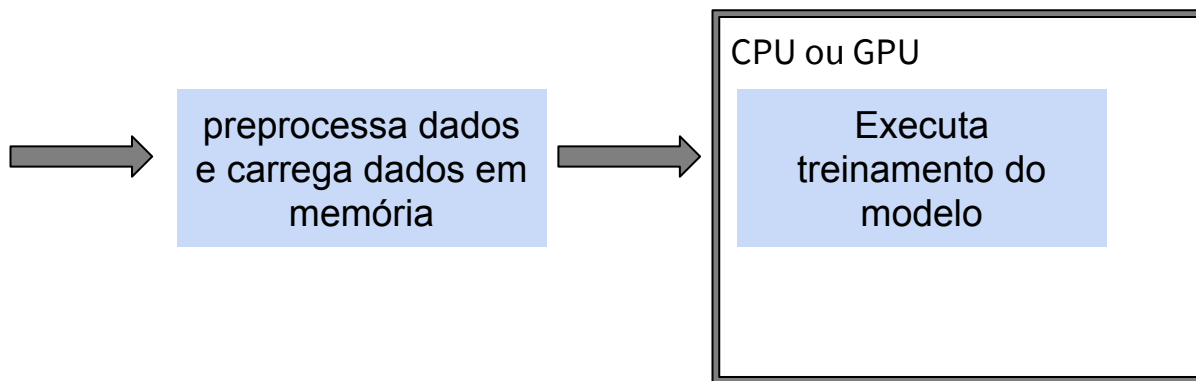
1. Coletar dados
2. Tratar e armazenar dados
3. Construir / modificar modelos
4. Treinar
5. Avaliar
6. Voltar para passo 3

Escalando o treinamento de um modelo

1. Coletar dados
2. Tratar e armazenar **dados**
3. Construir / modificar **modelos**
4. Treinar
5. Avaliar
6. Voltar para passo 3

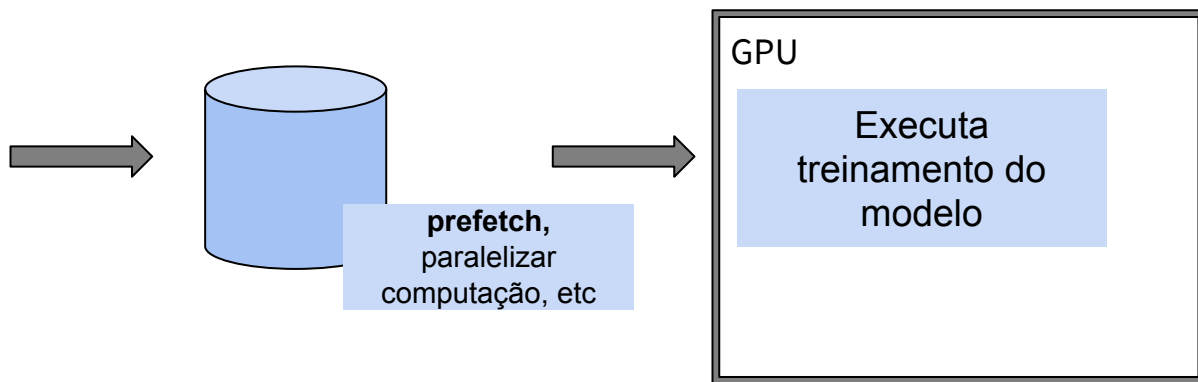
Escalando o treinamento de um modelo

1. Dados cabem em memória, modelo cabe em memória (configuração mais simples porém pouco comum)



Escalando o treinamento de um modelo

2. Dados não cabem em memória, modelo cabe em memória.



Quer saber mais?
[tf.data](#)
[performance](#)
[guide](#)

Prefetching

Antes...

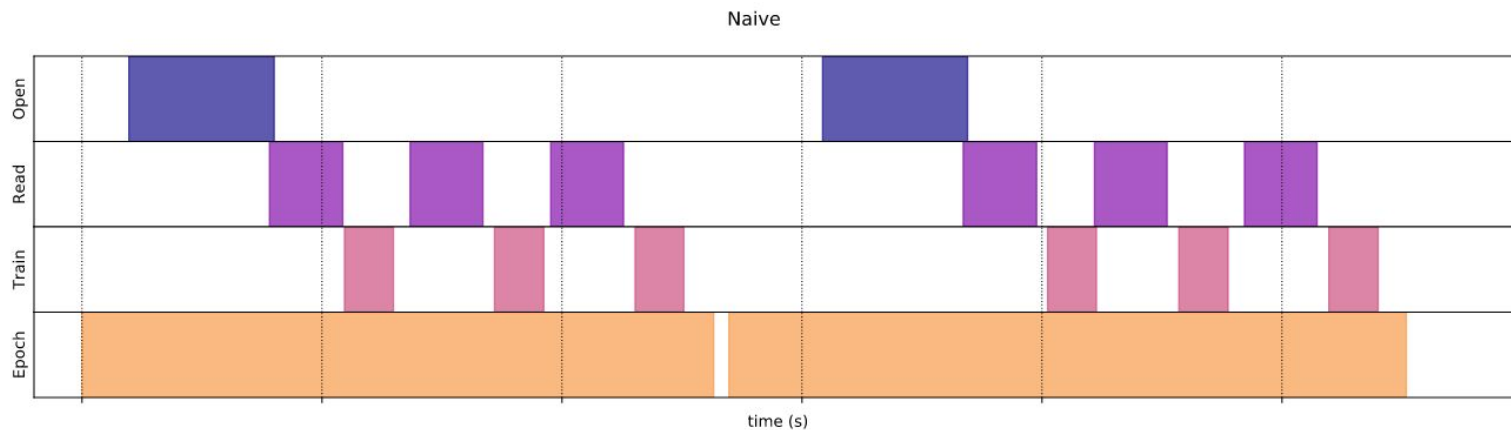


Imagem de https://www.tensorflow.org/guide/data_performance

Prefetching

Depois...

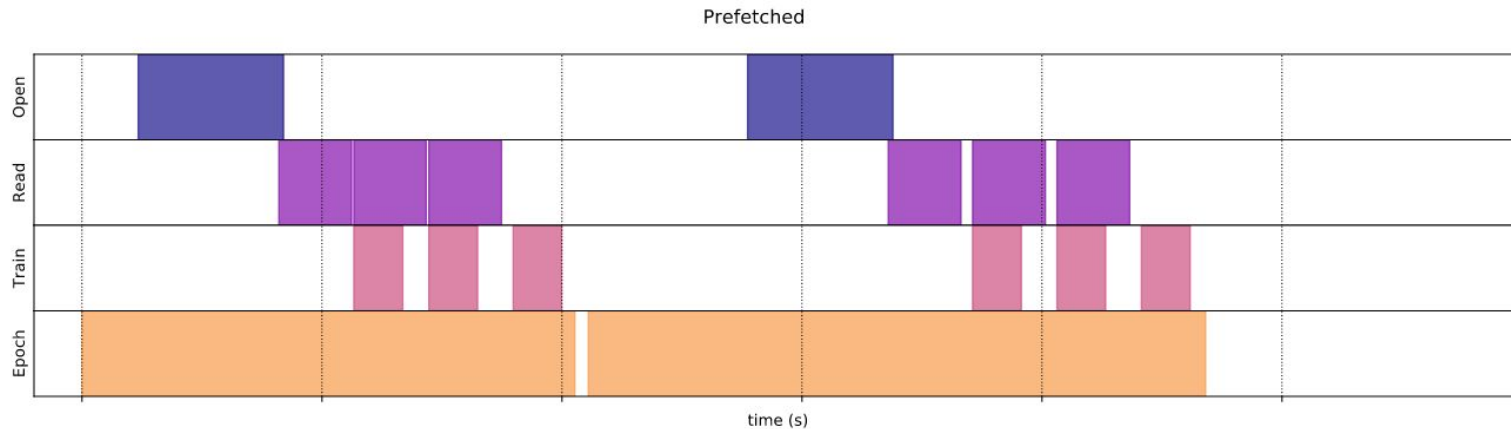
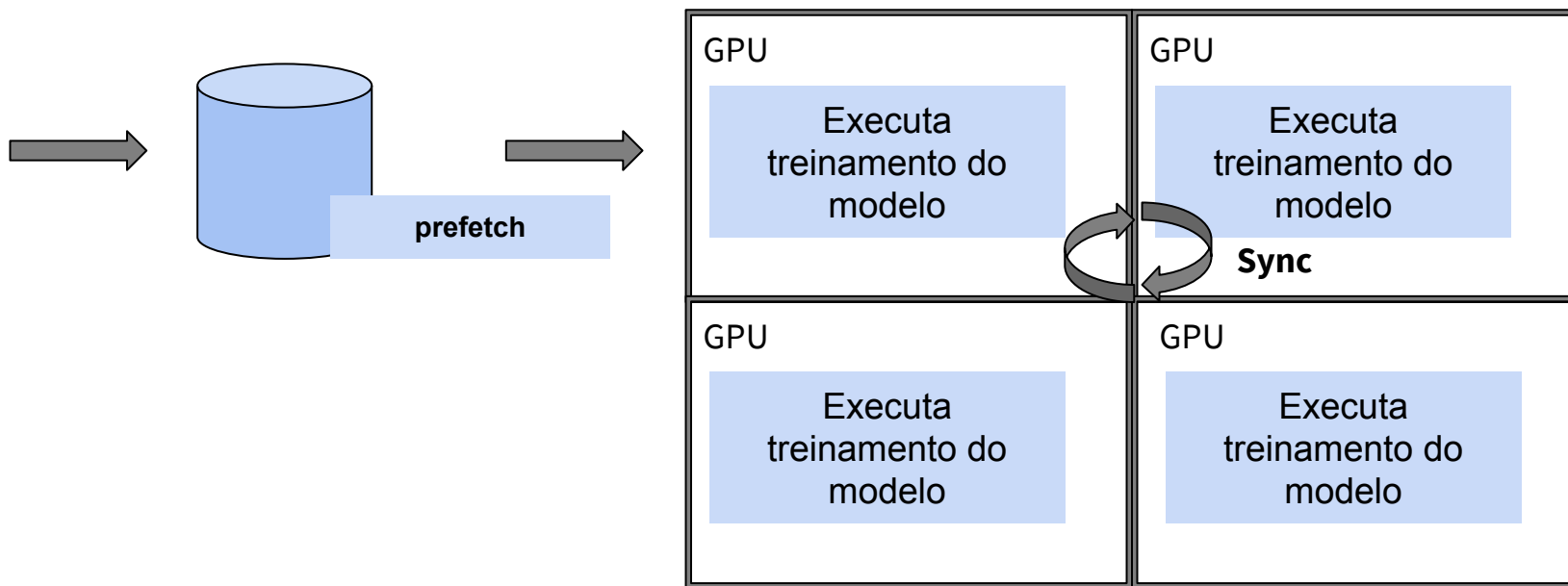


Imagem de https://www.tensorflow.org/guide/data_performance

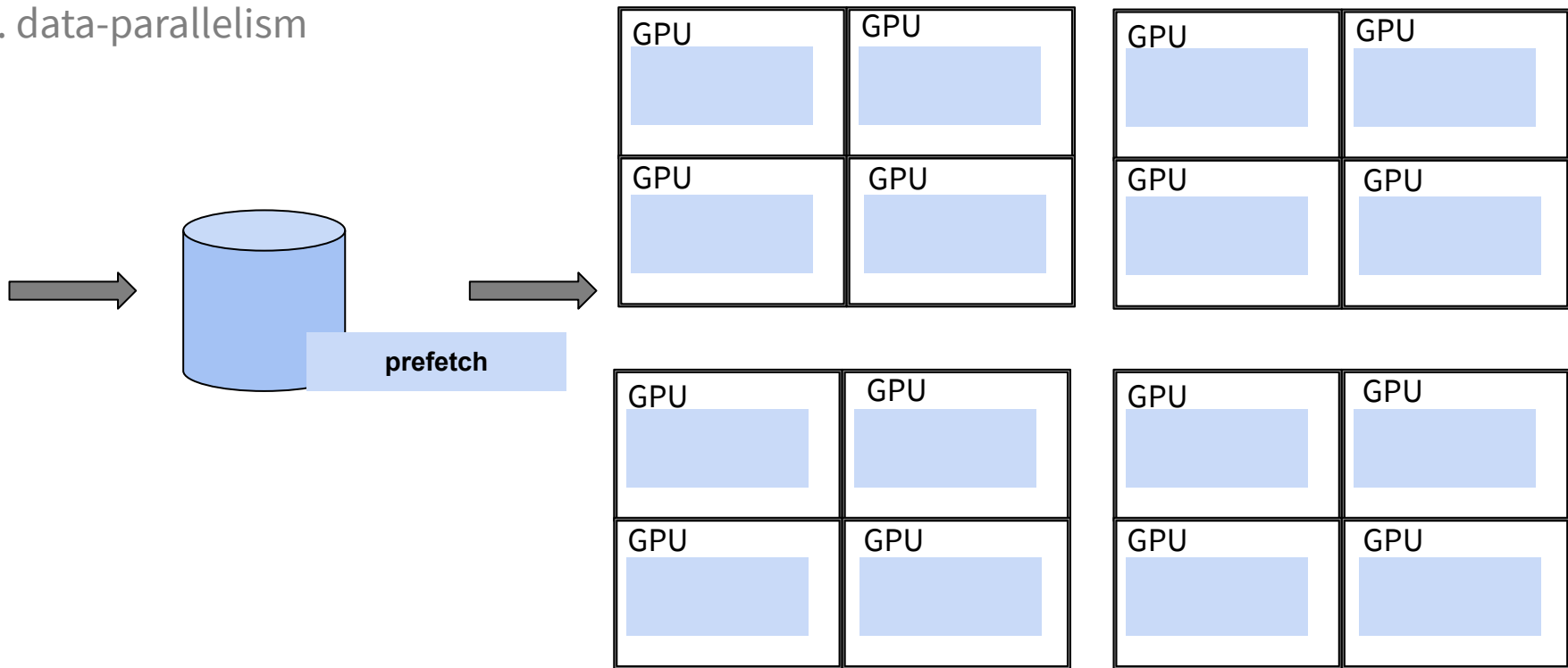
Escalando o treinamento de um modelo

3. multi-gpu



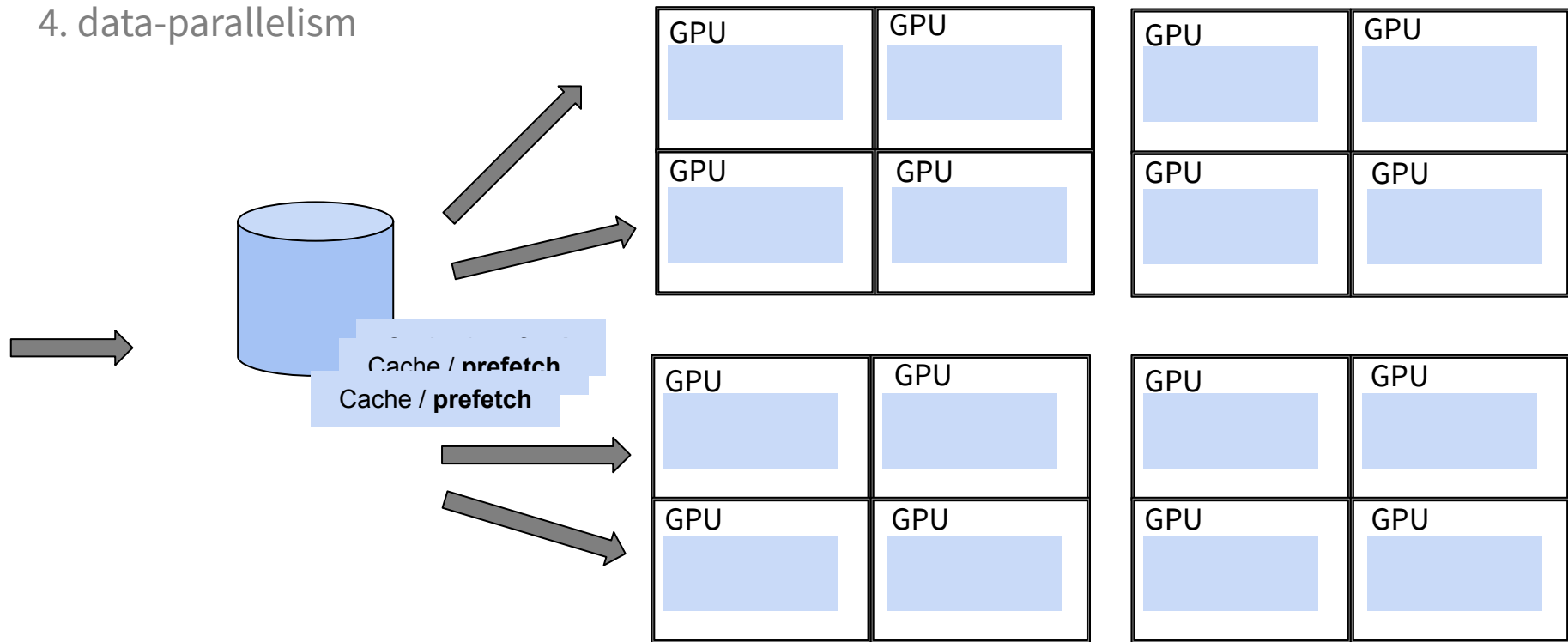
Escalando o treinamento de um modelo

4. data-parallelism



Escalando o treinamento de um modelo

4. data-parallelism



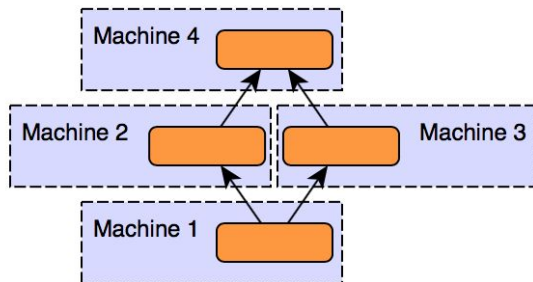
Escalando o treinamento de um modelo

5. model-parallelism: e se o modelo não couber em memória...

Escalando o treinamento de um modelo

5. model-parallelism: e se o modelo não couber em memória...

Model Parallelism



Data Parallelism

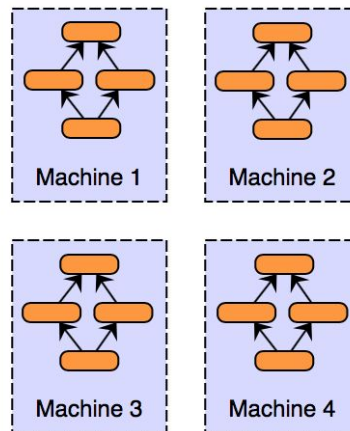


Imagem de

<https://xiandong79.github.io/Intro-Distributed-Deep-Learning>

Escalando o treinamento de um modelo

5. model-parallelism: e se o modelo não couber em memória...

Model and Data Parallelism

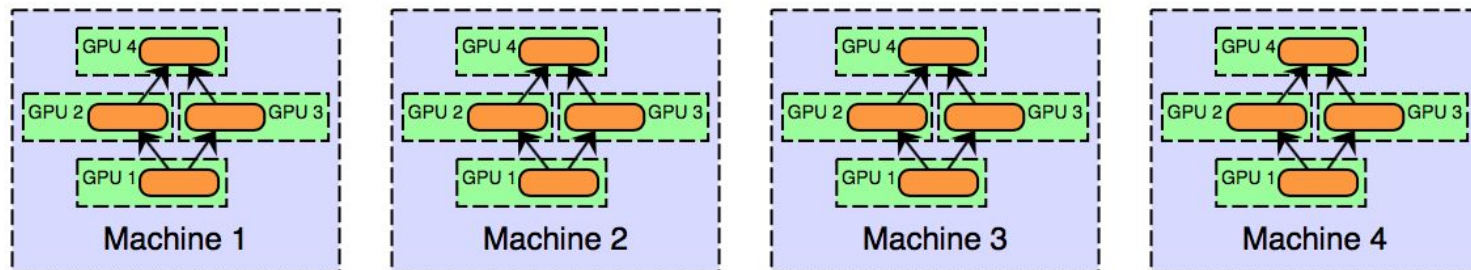


Imagem de

<https://xiandong79.github.io/Intro-Distributed-Deep-Learning>

Conclusões

Dados

**Treino e
Validação**

**Inferência
(Deploy)**

Como essas etapas se relacionam?

Treinamento

- Dados cabem em memória?
- Modelo cabe em memória?
- Quantas GPUs preciso pra treinar de forma eficiente?
- Quanto tempo posso levar pra treinar um modelo?
- Como preprocesso meus dados?
- Quais métricas são mais importantes?
 - Métricas são as mesmas durante inferência?
- ...

Características importantes

- Modelos falham silenciosamente.
 - Nada disso quebra nosso modelo: processamento diferente no treino e deploy, features trocadas, Features com valores errados, mudança na distribuição dos dados, ...
- Monitoramento é crucial, muitas vezes o melhor teste.
 - Crucial para entender: como o modelo se comporta, comparar modelos, identificar problemas e executar rollback, ...
- Seu modelo é tão bom quanto seus dados.
- Pesquisa != Produção.

Referências

1. [A Brief Guide to Running ML Systems in Production \(2020\)](#)
2. [tf.data: Build TensorFlow input pipelines](#)
3. [Better performance with the tf.data API](#)