



AULÃO 1 • ABERTURA DA FASE 2

0 motor novo

Hoje vocês vão ver por dentro o modelo que vão usar na Fase 2. Sem leitura prévia: é tudo na tela, agora.



COMO LER ESTE DOCUMENTO

Este PDF é o registro do Aulão 1. Cada página mostra uma tela do material interativo já aberta, num estado escolhido, e a faixa de baixo explica o que está acontecendo ali. Os laboratórios aparecem em mais de um estado, para vocês compararem o que muda quando um controle se mexe. O material de imersão completo, com todos os laboratórios para mexer, está no endereço da última página.

0 que muda da Fase 1 para cá

✓ tudo à vista

NA FASE 1

- O modelo começava **vazio**.
- Aprendia só com os dados de vocês.
- Dava para **desenhar no papel**.
- Se acertava, dava para explicar por quê.

NA FASE 2

- O modelo já vem **treinado por outra equipe**.
- Ele viu milhões de exemplos que vocês nunca verão.
- São **86 milhões de números** dentro dele.
- Vocês mexem em **quatro botões**, e só quatro.

É isso que deixa a Fase 2 possível com poucos dados. E é isso que traz problemas novos.

RESUMO

Na Fase 1 o modelo nascia vazio na mão de vocês. Aqui ele chega pronto, treinado por outra equipe, com 86 milhões de números ajustados por gente que vocês nunca vão conhecer. Vocês não vão mexer nesses 86 milhões: vão mexer em quatro botões.

Quatro blocos para vocês estudarem

Hoje eu abro os quatro na tela. Depois do aulão vocês refazem tudo com calma no material.

01

Ética

O modelo é emprestado. O que ele traz de bagagem, e o que vocês precisam escrever no relatório por causa disso.

02

A rede por dentro

O que é um neurônio, o que é uma camada, e o que acontece quando vocês empilham várias. Vocês vão montar uma rede aqui na tela.

03

Como ela aprende

Como a rede descobre que errou e se corrige. É aqui que fica o botão mais importante da fase inteira.

04

A sala de controle

As peças que já vêm prontas no caderno de código, e os três jeitos mais comuns de perder uma tarde de trabalho.



Errar aqui é de graça. Errar no desafio custa nota. Mexam em tudo.

Os quatro blocos do material de imersão. Neste aulão cada um virou um ou dois laboratórios na tela. Depois disso, o caminho é abrir o material e refazer tudo com calma, mexendo nos controles sem pressa.

0 modelo que vocês vão usar não é de vocês

Alguém treinou esse modelo antes de vocês, em outro país, com material que ninguém aqui escolheu. Isso cria três problemas que vocês herdam junto.

01

Ele herda o que estava torto

Se o material original tinha mais de um tipo de coisa e menos de outro, o modelo chega com essa inclinação. **O treino de vocês por cima não apaga isso.**

02

Ele erra com confiança

Um modelo diz **"97% de certeza"** mesmo quando está errado. Esse número é uma conta interna, e ela pode estar muito enganada.

03

Ele acerta pelo motivo errado

Dá para ir muito bem olhando para o lugar errado da foto. **A métrica não mostra isso.** No Capítulo 3 tem um laboratório inteiro sobre esse caso.

Nenhum dos três aparece na métrica. Todos os três aparecem no relatório, e o relatório **vale nota**.

Os três problemas nascem do mesmo fato: alguém treinou o modelo antes, com material que vocês não escolheram. Nenhum dos três aparece na métrica de acerto. Todos os três precisam aparecer no relatório de vocês.

0 modelo emprestado

O modelo aprendeu com fotos de um lugar onde a luz é boa. Agora vai trabalhar em outro, de luz fraca e celular antigo.

fotos de treino vindas do lugar a **90%**



Lugar A: luz boa, câmera boa.
Lugar B: luz fraca, celular antigo.

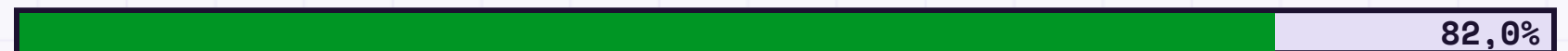
MÉTRICA ÚNICA DO RELATÓRIO

78,5%

parece um ótimo resultado

GRUPO A · LUZ BOA

aceitável



GRUPO B · LUZ FRACA

reprovado



UM GRUPO INTEIRO FOI ABANDONADO

Diferença de **35 pontos** entre A e B. Quem for atendido pelo lugar B recebe um sistema que erra mais da metade das vezes, e a métrica única não conta isso.

A média engana. O número geral fica bonito enquanto um grupo inteiro vai mal.

Controle em 90%: quase todas as fotos de treino vieram do lugar de luz boa. O grupo A vai a 82% e o grupo B despenca para 47%. A média geral marca 78,5% e parece um bom resultado — é ela que esconde o problema. Quem for atendido pelo lugar B recebe um sistema que erra mais da metade das vezes.

0 modelo emprestado

O modelo aprendeu com fotos de um lugar onde a luz é boa. Agora vai trabalhar em outro, de luz fraca e celular antigo.

fotos de treino vindas do
lugar a **55%**



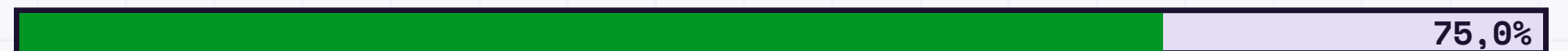
Lugar A: luz boa, câmera boa.
Lugar B: luz fraca, celular antigo.

MÉTRICA ÚNICA DO RELATÓRIO

72,3%

parece um bom resultado

GRUPO A · LUZ BOA



GRUPO B · LUZ FRACA



OS DOIS GRUPOS ANDAM JUNTOS

O treino viu os dois lugares em quantidade parecida. Nenhum grupo fica para trás, e a média diz a verdade.

A média engana. O número geral fica bonito enquanto um grupo inteiro vai mal.

Mesmo modelo, mesma conta, origem equilibrada em 55%. Agora os dois grupos andam juntos e a média diz a verdade. A diferença entre os dois estados não está no modelo: está em de onde vieram as fotos. Por isso o relatório precisa falar da origem dos dados, e por isso vale medir o desempenho por grupo, não só a média.

Confiar ou desconfiar?

Um assistente devolve a resposta **mais provável**, que não é a mesma coisa que a certa. Podem usar IA generativa nesta fase, desde que saibam explicar o que ela entregou e digam isso no relatório.

Caso 1

Caso 2

Caso 3

1 de 3 abertos

PERGUNTA DA EQUIPE · CASO 1

Qual taxa de aprendizado eu uso para ajustar o EfficientNet-B0 nesta fase?

RESPOSTA DA IA

Use **0.01**. É o valor padrão e funciona bem na maioria dos casos de classificação de imagens.

× PRECISA CONFERIR

O número tem cara de padrão porque é o padrão de treinar do zero. Modelo já treinado precisa de passo mil vezes menor. Nesta fase a faixa vai de 0,00001 a 0,02, e a ficha de cada modelo diz qual. Com 0,01 vocês estragam o que ele já sabia na primeira época.

✓ a turma acertou o veredito

A resposta parece segura e vem com um número redondo. O problema é que 0.01 é o padrão de quem treina do zero. Um modelo que já vem treinado precisa de um passo mil vezes menor, e a ficha de cada modelo diz qual. Aceitar esse valor estragaria o treino na primeira época.

Confiar ou desconfiar?

Um assistente devolve a resposta **mais provável**, que não é a mesma coisa que a certa. Podem usar IA generativa nesta fase, desde que saibam explicar o que ela entregou e digam isso no relatório.

Caso 1

Caso 2

Caso 3

2 de 3 abertos

PERGUNTA DA EQUIPE · CASO 2

O que o dropout de 0.3 faz na camada que está no nosso código?

RESPOSTA DA IA

A cada passagem de treino, 30% dos neurônios daquela camada são desligados ao acaso, para o modelo não depender demais de nenhum deles. Na validação todos voltam a ligar.

✓ DÁ PARA USAR, E DÁ PARA CONFERIR

É explicação conceitual, e bate com o que vocês viram no Laboratório 4. Conferir custa um minuto: leiam a documentação da camada. Esse é o uso que o relatório aceita sem problema: vocês entenderam e checaram.

✓ a turma acertou o veredito

Aqui a resposta está certa e é fácil de conferir: bate com o que a turma viu no Laboratório 4 e a documentação confirma em um minuto. Este é o uso que o relatório aceita sem problema, porque a equipe entendeu e checkou.

Confiar ou desconfiar?

Um assistente devolve a resposta **mais provável**, que não é a mesma coisa que a certa. Podem usar IA generativa nesta fase, desde que saibam explicar o que ela entregou e digam isso no relatório.

Caso 1

Caso 2

Caso 3

3 de 3 abertos

PERGUNTA DA EQUIPE · CASO 3

Escreve o bloco de treino inteiro para mim, com tudo pronto.

RESPOSTA DA IA

Claro! Segue o código completo de treino, já com otimizador, agendador e laço de épocas configurados.

× A PERGUNTA ERRADA

Aqui não é sobre a resposta estar certa. Se ninguém da equipe sabe explicar linha por linha o que colaram, o ponto vai embora na avaliação, e a equipe passa a fase inteira sem aprender o que a fase pede.

× o veredito era o outro

Este caso não é sobre a resposta estar certa. É sobre quem vai explicar o código na avaliação. Se ninguém da equipe sabe dizer o que cada linha faz, o ponto vai embora — e a equipe atravessa a fase sem aprender o que a fase pede.

Um neurônio é só uma conta

Ele pega os números que entram, multiplica cada um por um **peso**, soma tudo e passa o resultado por uma função. É essa peça que se repete milhões de vezes dentro do modelo.

entrada x_1 0,80

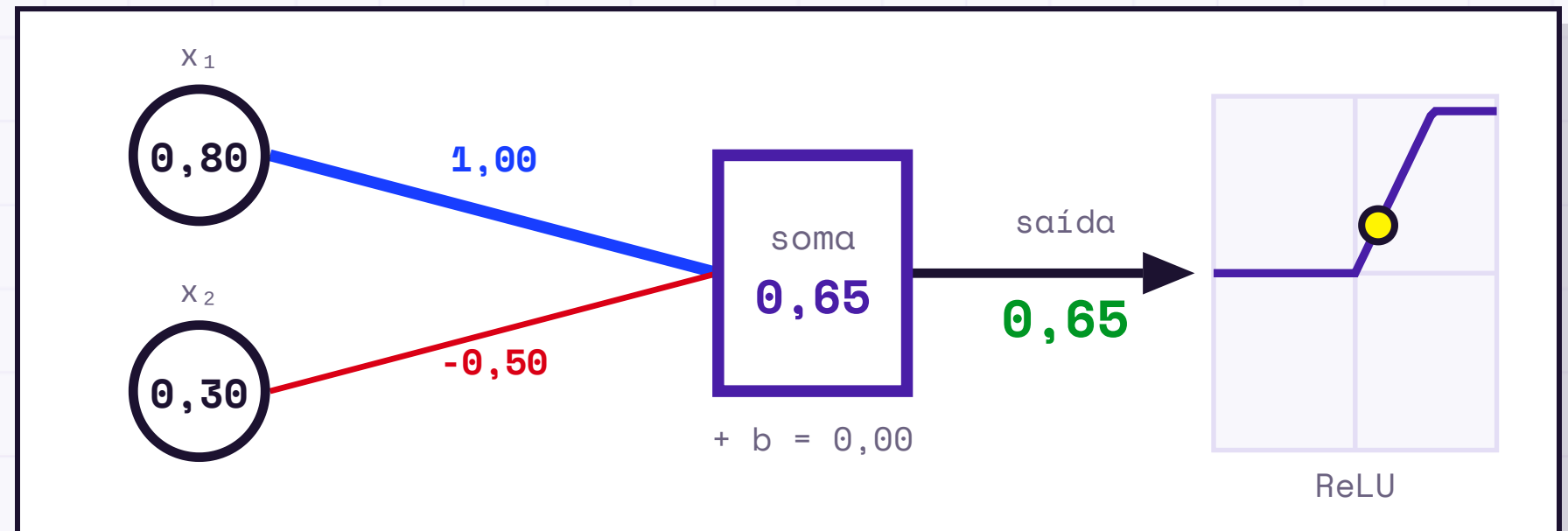
entrada x_2 0,30

peso w_1 1,00

peso w_2 -0,50

deslocamento b 0,00

FUNÇÃO DE ATIVAÇÃO



A CONTA, POR EXTENSO

$$z = (0,80 \times 1,00) + (0,30 \times -0,50) + 0,00 = 0,65$$

$$\text{saída} = \text{ReLU}(0,65) = 0,65$$

A **ReLU** zera tudo que é negativo e deixa passar o resto. Mexam em **b** até a saída travar em 0: é a ReLU desligando o neurônio.

A conta inteira de um neurônio, com os valores da tela: cada entrada é multiplicada pelo seu peso, tudo é somado, o deslocamento entra, e o resultado passa pela função de ativação. Com ReLU, o que é negativo vira zero e o resto passa. Repare que é só isso — e essa peça se repete milhões de vezes dentro do modelo.

Um neurônio é só uma conta

Ele pega os números que entram, multiplica cada um por um **peso**, soma tudo e passa o resultado por uma função. É essa peça que se repete milhões de vezes dentro do modelo.

entrada x_1 0,80

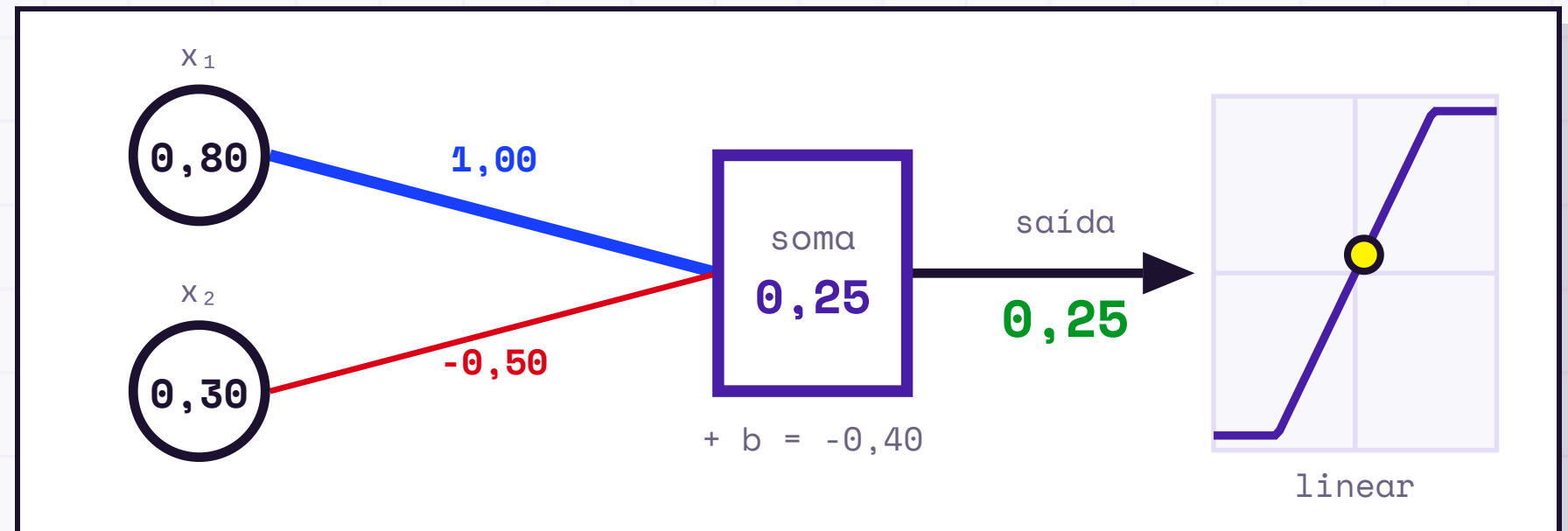
entrada x_2 0,30

peso w_1 1,00

peso w_2 -0,50

deslocamento b -0,40

FUNÇÃO DE ATIVAÇÃO



A CONTA, POR EXTENSO

$$z = (0,80 \times 1,00) + (0,30 \times -0,50) + -0,40 = 0,25$$

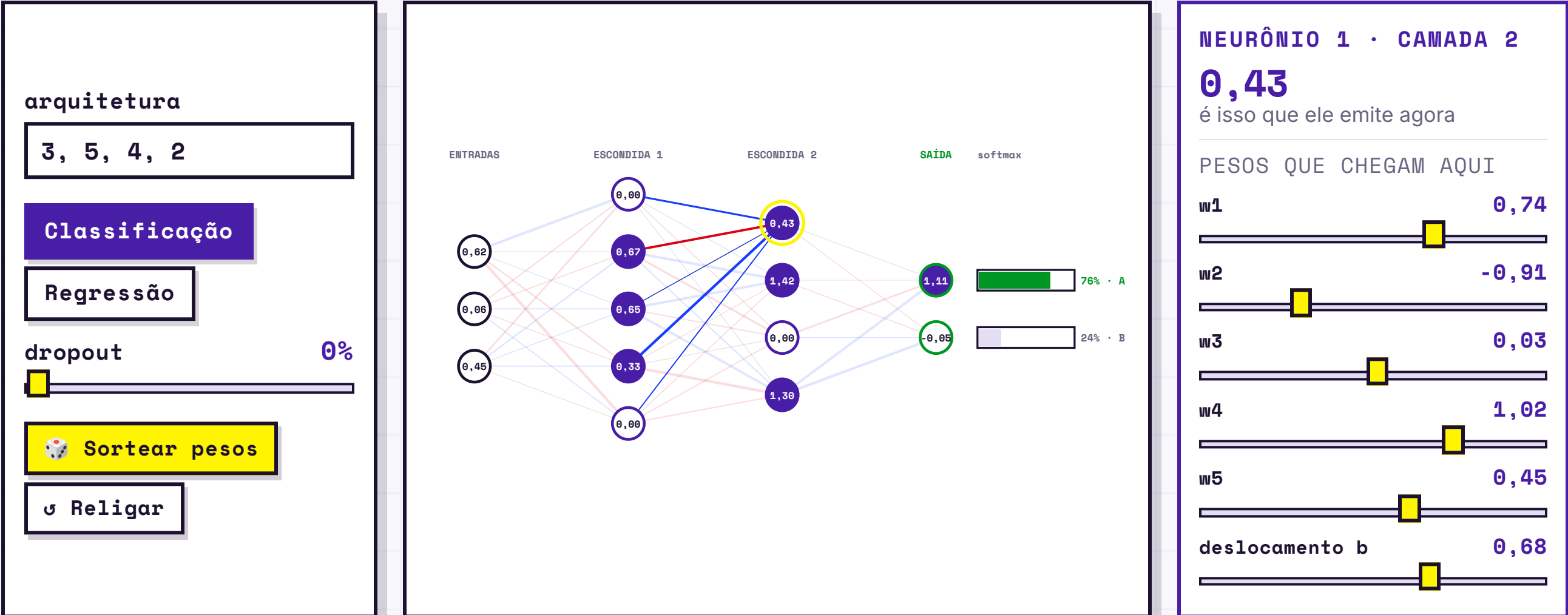
$$\text{saída} = \text{linear}(0,25) = 0,25$$

Sem ativação a saída é **uma reta**. Duas camadas lineares empilhadas dão exatamente o mesmo resultado que uma só: você paga o dobro e não ganha nada.

Agora sem ativação nenhuma. A saída vira uma reta, e aqui está o motivo de a ativação existir: duas camadas lineares empilhadas dão exatamente o mesmo resultado que uma só. Sem ela, empilhar camadas custa o dobro e não ganha nada. É a ativação que dobra a linha e faz a profundidade valer a pena.

Agora a rede inteira

3, 5, 4, 2 é 3 entradas, camadas de 5 e de 4, e 2 saídas. Cliquem num círculo para mexer nos pesos dele.



Rede 3, 5, 4, 2: três entradas, duas camadas escondidas e duas saídas. O neurônio destacado está emitindo o número que aparece no painel da direita, e os controles mostram os pesos que chegam nele. À direita da rede, o softmax transformou as duas saídas em porcentagens que somam 100%: a classe escolhida é a de maior barra.

Agora a rede inteira

3, 5, 4, 2 é 3 entradas, camadas de 5 e de 4, e 2 saídas. Cliquem num círculo para mexer nos pesos dele.

arquitetura

3, 5, 4, 2

Classificação

Regressão

dropout

35%

Sortear pesos

Religar

ENTRADAS

ESCONDIDA 1

ESCONDIDA 2

SAÍDA softmax

NEURÔNIO 2 · CAMADA 2

0,79

é isso que ele emite agora

PESOS QUE CHEGAM AQUI

w1	0,19
w2	0,99
w3	0,98
w4	-0,31
w5	-0,40
deslocamento b	0,23

LABORATÓRIO 4 · COM DROPOUT

Mesma rede, dropout em 35%. Os neurônios marcados com X foram desligados ao acaso nesta passagem, e as ligações que saíam deles ficaram apagadas. A cada passagem o sorteio muda. É assim que o modelo é obrigado a espalhar o que aprendeu em vez de depender de um neurônio só. Isso acontece apenas no treino: na validação todos voltam a ligar.

Agora a rede inteira

3, 5, 4, 2 é 3 entradas, camadas de 5 e de 4, e 2 saídas. Cliquem num círculo para mexer nos pesos dele.

arquitetura

3, 5, 4, 2

Classificação

Regressão

dropout

0%

🎲 Sortear pesos

⏮ Religar

ENTRADAS ESCONDIDA 1 ESCONDIDA 2 SAÍDA

0,11 0,38 0,35 0,00 0,42 0,68 0,64 0,58 1,23 0,88 0,00 0,56 **0,51**

o número já é a re

NEURÔNIO DE SAÍDA 1

0,51

é isso que ele emite agora

PESOS QUE CHEGAM AQUI

w1 0,02

w2 -0,19

w3 -0,74

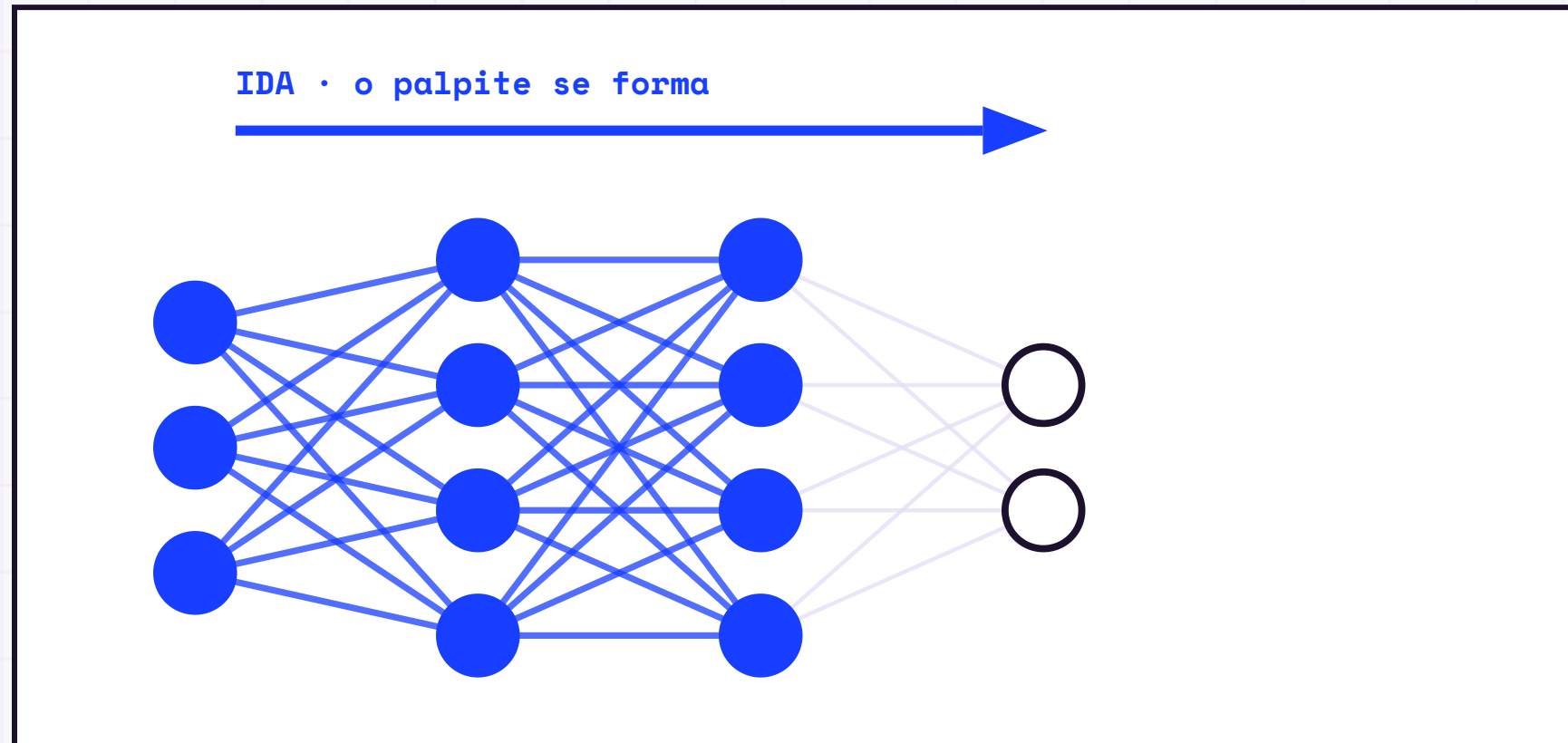
w4 0,98

deslocamento b 0,10

Trocando a tarefa para regressão, a última camada passa a ter um neurônio só e o softmax desaparece. O número que sai já é a resposta: o preço de um imóvel, a temperatura de amanhã. A tarefa decide como a rede termina.

Como a rede descobre que errou

São cinco passos que se repetem milhares de vezes por treino. O nome disso é **backpropagation**: propagar o erro para trás.



► **Tocar um passo de treino**

→ **Avançar um quadro**

↺ **Recomeçar**

1 · A informação entra

2 · Ela atravessa a rede

3 · Sai um palpite

4 · O erro volta para trás

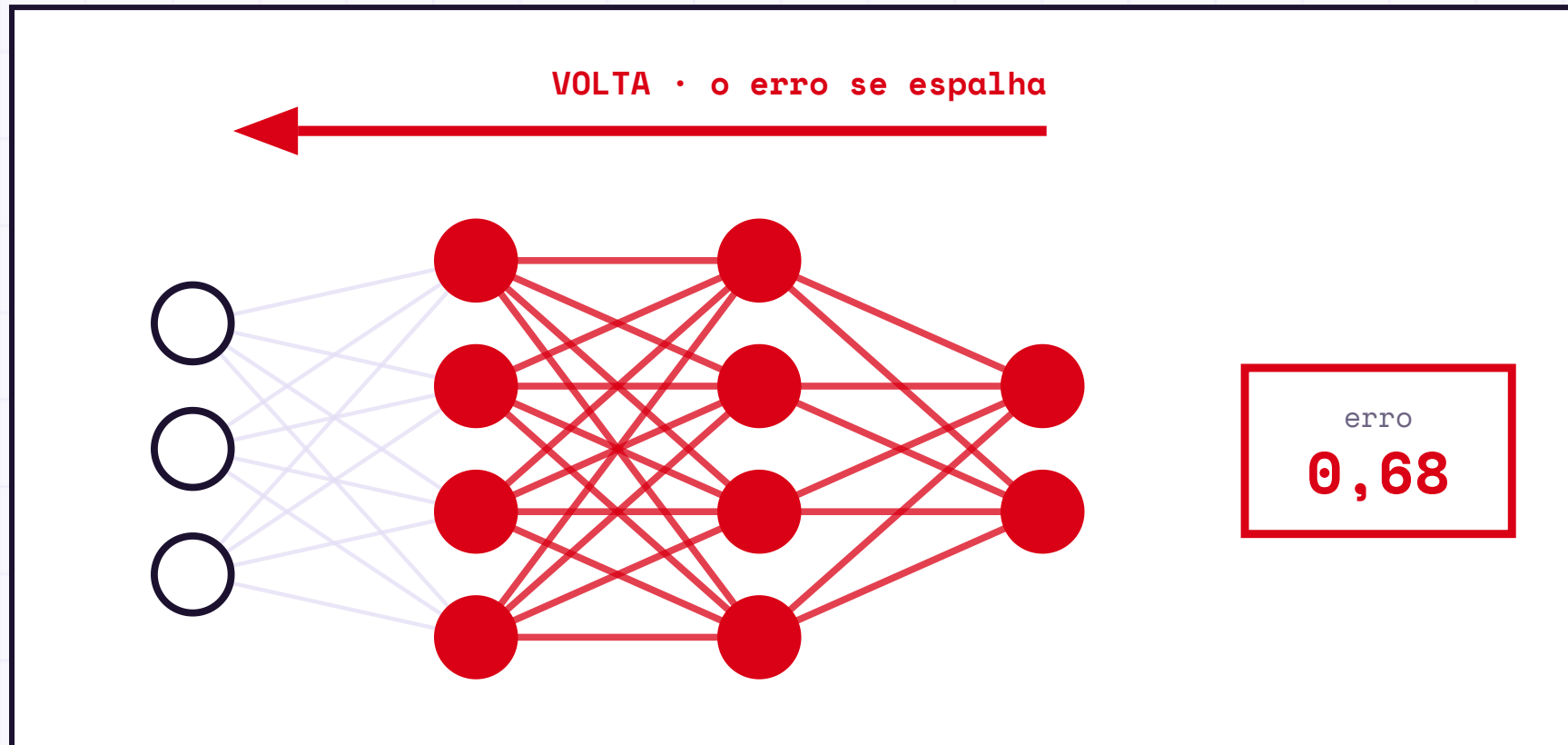
5 · Os pesos mudam

Quadro 2 de 5. Cada neurônio faz a conta e passa o resultado adiante.

O exemplo entrou e está atravessando a rede, camada por camada. Cada neurônio faz a conta que vocês viram no Laboratório 3 e passa o resultado adiante. No fim desse caminho sai um palpite. Nada foi corrigido ainda.

Como a rede descobre que errou

São cinco passos que se repetem milhares de vezes por treino. O nome disso é **backpropagation**: propagar o erro para trás.



► **Tocar um passo de treino**

→ **Avançar um quadro**

↺ **Recomeçar**

1 · A informação entra

2 · Ela atravessa a rede

3 · Sai um palpite

4 · O erro volta para trás

5 · Os pesos mudam

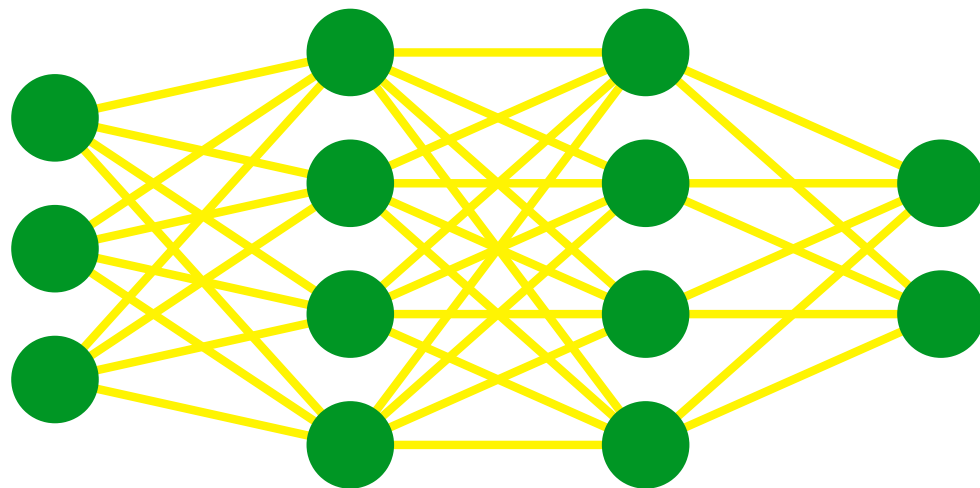
Quadro 4 de 5. A rede descobre quanto cada peso contribuiu para o erro.

O palpite foi comparado com a resposta certa e nasceu um número: o erro, 0,68. Agora ele volta pela rede no sentido contrário. Camada por camada, a rede descobre o quanto cada peso contribuiu para esse erro — e, portanto, para que lado cada um precisa ir.

Como a rede descobre que errou

São cinco passos que se repetem milhares de vezes por treino. O nome disso é **backpropagation**: propagar o erro para trás.

TODOS OS PESOS SE AJUSTAM



erro menor
0,41

- 1 · A informação entra
- 2 · Ela atravessa a rede
- 3 · Sai um palpite
- 4 · O erro volta para trás
- 5 · Os pesos mudam

► **Tocar um passo de treino**

→ **Avançar um quadro**

↺ **Recomeçar**

A rede descobriu a **direção** de cada peso. O tamanho do passo é outra decisão, e é o próximo laboratório.

Os pesos andam na direção que diminui o erro, e ele cai de 0,68 para 0,41. Repare no que o backpropagation entrega: a direção de cada peso, e só isso. O tamanho do passo é outra decisão, e é o assunto do próximo laboratório.

A taxa de aprendizado

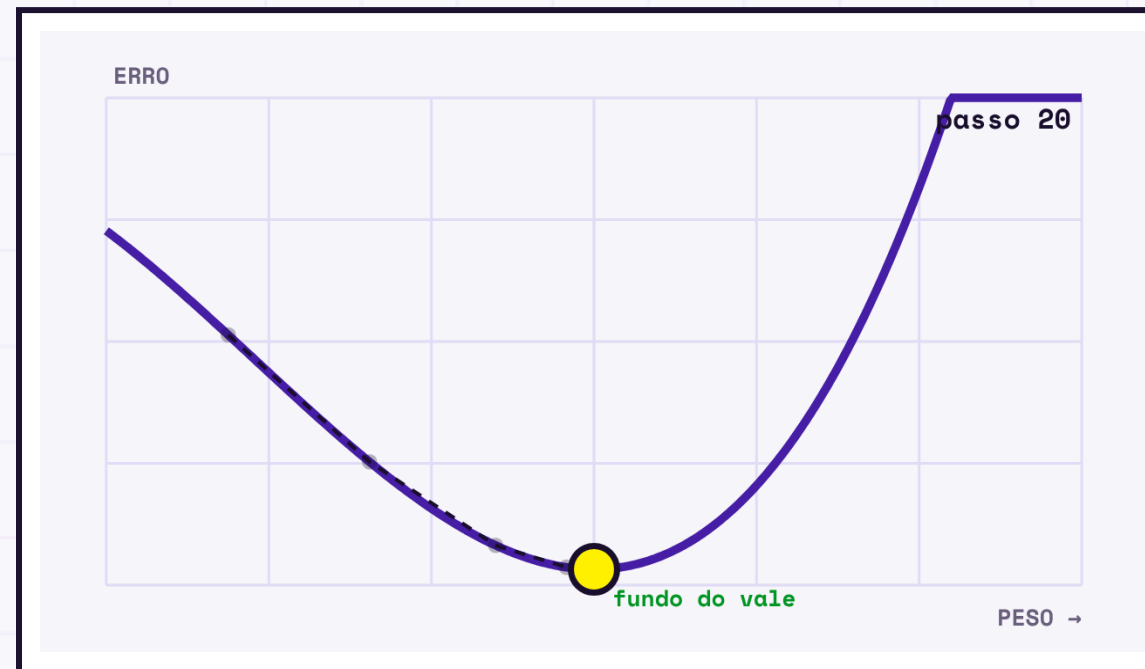
Imaginem descer uma montanha no escuro. O passo anterior diz para que lado o chão desce. A **taxa** é o tamanho do passo que vocês dão, e as **épocas** são quantos passos cabem no tempo.

taxa de aprendizado **0,35**

épocas disponíveis **20**

onde começar **-2,4**

Treinar **Reiniciar**



ERRO NO FIM

0,30

CHEGOU AO FUNDO

Passo e épocas combinaram. **Passo pequeno exige muitas épocas.**

Guardem este número. Nos modelos desta fase a taxa vai de 0,00001 a 0,02. São mil vezes de diferença: modelo grande e já treinado precisa de passo minúsculo para não estragar o que já sabia.

Taxa 0,35 com 20 épocas. A bolinha desce em passos que encurtam sozinhos conforme o chão fica plano, e chega ao fundo do vale com sobra de épocas. É o comportamento que vocês querem ver: a perda cai e assenta.

A taxa de aprendizado

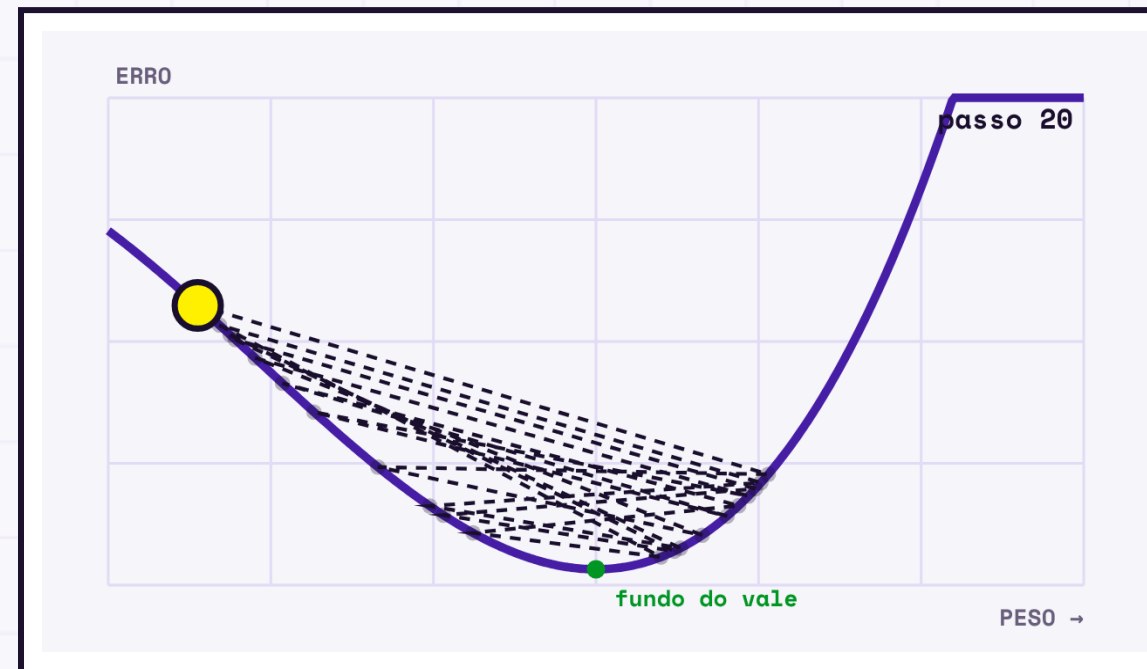
Imaginem descer uma montanha no escuro. O passo anterior diz para que lado o chão desce. A **taxa** é o tamanho do passo que vocês dão, e as **épocas** são quantos passos cabem no tempo.

taxa de aprendizado **1,10**

épocas disponíveis **20**

onde começar **-2,4**

Treinar **Reiniciar**



ERRO NO FIM

5,28**ESTÁ QUICANDO**

A bolinha pula de um lado ao outro sem descer. Diminuem a taxa.

Guardem este número. Nos modelos desta fase a taxa vai de 0,00001 a 0,02. São mil vezes de diferença: modelo grande e já treinado precisa de passo minúsculo para não estragar o que já sabia.

Mesma montanha, taxa 1,10. Cada passo passa do ponto e cai do outro lado do vale. A bolinha fica quicando e o erro no fim é pior do que no começo. No caderno de vocês isso aparece como uma perda que sobe, oscila, ou vira NaN. O remédio é dividir a taxa.

A taxa de aprendizado

Imaginem descer uma montanha no escuro. O passo anterior diz para que lado o chão desce. A **taxa** é o tamanho do passo que vocês dão, e as **épocas** são quantos passos cabem no tempo.

taxa de aprendizado **0,03**

épocas disponíveis **20**

onde começar **-2,4**

Treinar **Reiniciar**



ERRO NO FIM

1,24

PAROU NO MEIO DO CAMINHO

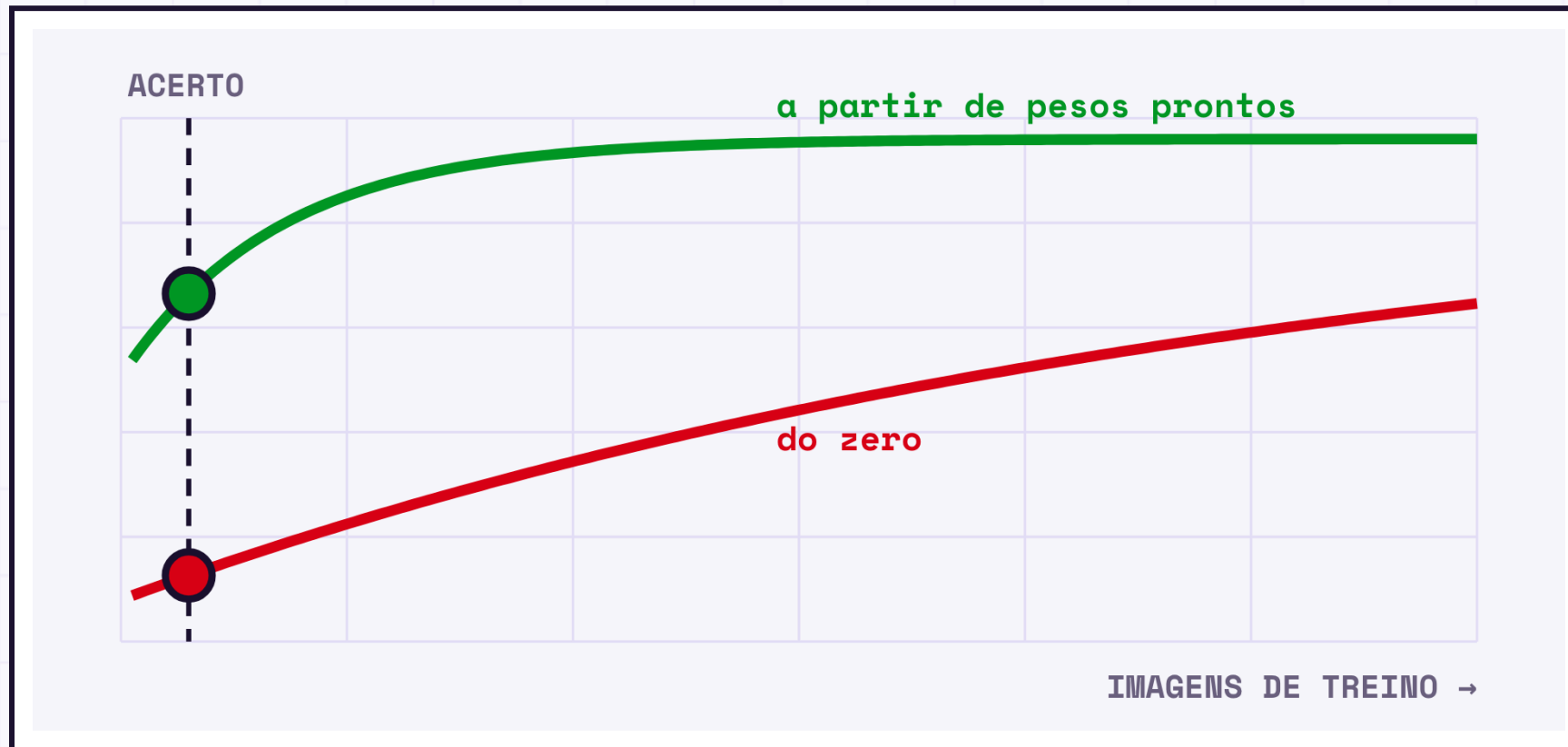
Acabaram as épocas antes do fundo. Aumentem as épocas ou o passo.

Guardem este número. Nos modelos desta fase a taxa vai de 0,00001 a 0,02. São mil vezes de diferença: modelo grande e já treinado precisa de passo minúsculo para não estragar o que já sabia.

Agora taxa 0,03. Os passos são tão curtos que as épocas acabam antes de o modelo sair do lugar. O treino termina sem erro nenhum na tela e sem ter aprendido quase nada. Guardem a faixa desta fase: de 0,00001 a 0,02, mil vezes de diferença entre um modelo e outro.

Por que isso funciona com poucas fotos

O modelo já olhou mais de um milhão de fotos antes de conhecer vocês. Ele já sabe o que é borda, textura e brilho: falta aprender só as classes do problema de vocês.



imagens de treino **300**

A PARTIR DE PESOS PRONTOS

69,9%

falta ensinar só as classes de vocês

DO ZERO

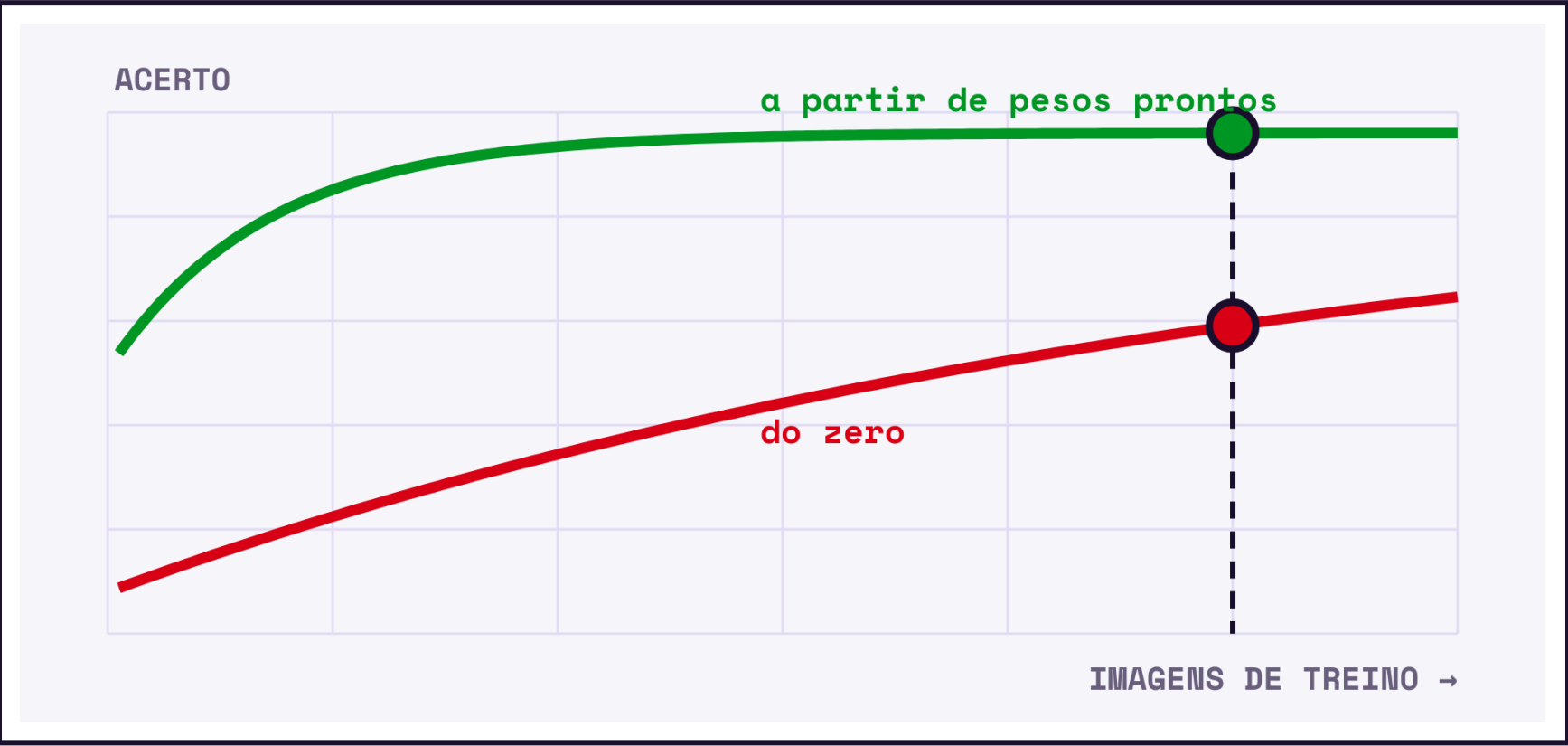
29,5%

precisa aprender a enxergar antes

Com 300 imagens, partir de pesos prontos entrega 69,9% e treinar do zero entrega 29,5%. Quarenta pontos de diferença. O modelo pronto já sabe o que é borda, textura e brilho: falta ensinar só as classes do problema de vocês. O do zero precisa aprender a enxergar primeiro, e com 300 fotos não dá.

Por que isso funciona com poucas fotos

O modelo já olhou mais de um milhão de fotos antes de conhecer vocês. Ele já sabe o que é borda, textura e brilho: falta aprender só as classes do problema de vocês.



imagens de treino

5.000

A PARTIR DE PESOS PRONTOS

92,0%

falta ensinar só as classes de vocês

DO ZERO

64,3%

precisa aprender a enxergar antes

Com 5.000 imagens o treino do zero começa a alcançar, e a distância encolhe. Só que nesta fase vocês não terão esse tanto de dados. É exatamente por isso que a Fase 2 parte de um modelo já treinado: é o que torna o problema resolvível com o que vocês têm.

Quatro peças que já vêm prontas

Ninguém precisa escrever essas quatro: elas já vêm no caderno de código. Mas quando o treino der errado, o problema vai estar em uma delas. Cliquem para ver o que cada uma faz.

OTIMIZADOR

Como cada peso muda

Decide como cada peso muda depois que o backpropagation descobriu a direção. O **Adam** guarda memória dos passos anteriores: acelera onde a descida é consistente e freia onde o caminho fica irregular.

Um anda de olhos vendados contando passos. O outro lembra

AGENDADOR

O passo vai encurtando

Diminui o tamanho do passo ao longo do treino. Começa grande, para cobrir distância, e vai encurtando, para não passar do ponto no fim.

Você corre até chegar perto de casa e depois anda devagar para achar a porta.

PARADA ANTECIPADA

Para quando não melhora

Vigia a validação. Quando ela para de melhorar por algumas épocas seguidas, o treino é interrompido, mesmo que ainda houvesse épocas de sobra.

Continuar dali em diante só faz o modelo decorar.

PONTO DE VERIFICAÇÃO

Guarda a melhor época

Salva uma cópia do modelo na melhor época, não na última. Sem ele, a parada antecipada entrega o modelo de quando já estava piorando.

É o único que protege contra perder a tarde inteira.

As quatro peças já vêm escritas no caderno de código: ninguém precisa criá-las. O trabalho de vocês é reconhecer cada uma quando o treino der errado. O ponto de verificação é o único que protege contra perder a tarde inteira, e é o que o próximo laboratório mostra na prática.

Os três jeitos de perder uma tarde

O código roda num computador da nuvem, com placa de vídeo. Três coisas dão errado ali, sempre. Vamos cometer as três de propósito, agora.

Ambiente: CPU

Ambiente: GPU

✨ A sessão caiu

💾 Salvar ponto de verificação

ÚLTIMO PONTO SALVO
nenhum

TEMPO POR ÉPOCA

4 min

20 épocas × 4 min = 1 h 20 de espera

ÉPOCAS CONCLUÍDAS

3

de 20 planejadas

15%

× ERRO Nº 1 · AMBIENTE ERRADO

Erro nº 1: **esquecer de ligar a GPU**. O código roda igual, sem mensagem de erro nenhuma. Só demora vinte vezes mais. Confirmam o tipo de ambiente antes de apertar treinar.

Erro número um: esquecer de ligar a placa de vídeo. O código roda exatamente igual, sem mensagem de erro nenhuma — só demora cerca de vinte vezes mais. Quatro minutos por época viram uma hora e vinte de espera. Confirmam o tipo de ambiente antes de apertar treinar.

Os três jeitos de perder uma tarde

O código roda num computador da nuvem, com placa de vídeo. Três coisas dão errado ali, sempre. Vamos cometer as três de propósito, agora.

Ambiente: CPU

Ambiente: GPU

⚡ A sessão caiu

💾 Salvar ponto de verificação

ÚLTIMO PONTO SALVO
nenhum

TEMPO POR ÉPOCA

11 s

20 épocas × 11 s = pouco mais de 3 minutos

ÉPOCAS CONCLUÍDAS

0

de 20 planejadas

0%

× ERRO Nº 2 · PERDERAM TUDO

Erro nº 2: a **sessão desliga sozinha** depois de um tempo sem uso, e leva tudo junto. Sem ponto de verificação vocês perderam 14 épocas e recomeçam do zero.

Com a GPU ligada o treino cabe no tempo do encontro. Mas a sessão da nuvem desliga sozinha depois de um tempo sem uso, e leva tudo junto. Sem ponto de verificação, as épocas já treinadas foram embora e o trabalho recomeça do zero.

Os três jeitos de perder uma tarde

O código roda num computador da nuvem, com placa de vídeo. Três coisas dão errado ali, sempre. Vamos cometer as três de propósito, agora.

Ambiente: CPU

Ambiente: GPU

⚡ A sessão caiu

💾 Salvar ponto de verificação

ÚLTIMO PONTO SALVO
época 10

TEMPO POR ÉPOCA

11 s

20 épocas × 11 s = pouco mais de 3 minutos

ÉPOCAS CONCLUÍDAS

10

de 20 planejadas

50%

A SESSÃO CAIU · MAS DEU PARA VOLTAR

A sessão caiu, mas o ponto de verificação estava salvo na época 10. Perderam só 6 épocas. Erro nº 3 é **deixar para salvar depois**. Salvem a cada época.

Agora com o ponto de verificação salvo antes da queda. A sessão cai do mesmo jeito, mas o treino volta da época guardada em vez do zero. O terceiro erro é deixar para salvar depois: salvem a cada época, não no fim.

Cinco perguntas

1

2

3

4

5

PERGUNTA 1 DE 5

O modelo já vem treinado por outra equipe. O que o treino de vocês por cima **NÃO** consegue apagar?

A A inclinação que ele herdou do material de treino original

B O nome do arquivo de pesos

C A arquitetura da última camada

D O tamanho do lote

✓ **ISSO MESMO**

Ele herda o que estava torto no material original, e o ajuste por cima não desfaz isso. Por isso o relatório precisa falar da origem dos pesos.

Cinco perguntas

1

2

3

4

5

PERGUNTA 2 DE 5

A métrica geral do modelo deu 78%. Por que isso pode não bastar no relatório?

A Porque 78% é sempre pouco

B Porque a média pode esconder um grupo inteiro indo mal

C Porque a métrica geral não existe em redes neurais

D Porque falta calcular a média de novo

✓ **ISSO MESMO**

Foi o Laboratório 1: a média fica bonita enquanto um grupo vai a 47%. Desempenho por grupo é o que revela.

Cinco perguntas

1

2

3

4

5

PERGUNTA 3 DE 5

O backpropagation descobre o quê, exatamente?

A O tamanho do passo que cada peso deve dar

B A direção para onde cada peso precisa ir

C A quantidade de épocas necessária

D Qual classe está certa

✓ **ISSO MESMO**

Ele descobre a direção. O tamanho do passo é a taxa de aprendizado, e essa é decisão de vocês.

Cinco perguntas

1

2

3

4

5

PERGUNTA 4 DE 5

Por que um modelo grande e já treinado precisa de uma taxa de aprendizado minúscula?

A Porque ele é lento

B Para não estragar o que ele já sabia

C Porque a placa de vídeo exige

D Para caber na memória

✓ ISSO MESMO

Passo grande num modelo já treinado destrói o que ele aprendeu com um milhão de fotos. Nesta fase a taxa vai de 0,00001 a 0,02.

Cinco perguntas

1

2

3

4

5

PERGUNTA 5 DE 5

A sessão caiu no meio do treino e não havia ponto de verificação. O que vocês perdem?

A Nada, o treino continua de onde parou

B Só a última época

C Tudo: o treino recomeça do zero

D Só os dados, o modelo fica

✓ **ISSO MESMO**

Sem ponto de verificação não existe cópia salva. Foi o último laboratório: salvem a cada época.

O QUE LEVAR DAQUI

Três coisas e um endereço

O modelo é emprestado, e isso vai para o relatório.

A taxa de aprendizado é o botão que decide o treino.

Sem ponto de verificação, uma sessão que cai leva tudo.

Agora abram o material de imersão e refaçam os laboratórios com calma.

`olimpiadadeia.ceia.digital`

`olimpiada@ceia.ufg.br`



PRÓXIMO PASSO

O material de imersão da Fase 2 tem os quatro blocos completos, com dezoito laboratórios para mexer. Abram o Capítulo 1 e refaçam tudo o que viram aqui, com calma e mudando os valores. É lá que estão as fichas de cada modelo, com os valores de partida que vocês vão usar no desafio.