Insper

Redes Neurais Artificiais e Deep Learning

Avaliação Final – AF . 2026.2

Exemplo

Prezado(a) Aluno(a),

Você terá 120 minutos a partir do início oficial da prova para concluir esta avaliação, administre bem o seu tempo. Leia atentamente as instruções a seguir e as questões da prova antes de começar a resolvê-la.

1. Identifique-se com letra legível nas duas últimas páginas da prova.
2. Esta avaliação é composta de 14 questões de múltipla escolha (0,5 ponto cada) e 1 questão dissertativa em três partes (3,0 pontos), totalizando 10,0 pontos.
3. Verifique se a prova está completa e/ou se há problemas de impressão e comunique o aplicador antes de iniciar a prova. Comunicação posterior não será considerada. A prova tem 13 páginas.
4. **Devolva ao aplicador apenas as duas últimas páginas (páginas 12 e 13)**, com o formulário de respostas e sua identificação – não erre o lançamento das respostas e não esqueça a identificação.
5. A resolução da prova poderá ser feita a lápis ou a caneta. Avaliações feitas a lápis, no entanto, não serão revisadas pelo professor.
6. Em caso de dúvida sobre alguma questão desta avaliação, redija um texto na folha de respostas ou seu verso explicitando-a para que o professor avalie a pertinência durante a correção.
7. Portar qualquer material estranho à folha de questões – celular (ligado ou desligado), tablet, notebook, anotações e livro – constituirá violações ao Código de Ética e de Conduta e acarretará sanções nele previstas.
8. Consulta a colegas e a qualquer material estranho à folha de questões constituirá violação ao Código de Ética e de Conduta e acarretará sanções nele previstas. Faça o seu trabalho de maneira ética!
9. Você somente poderá sair da sala depois de entregar a prova. Caso necessite sair durante a realização da avaliação, peça autorização antecipadamente ao aplicador.

Boa Prova!

Para uso exclusivo do Professor

Orientações durante a avaliação:

Permitido utilizar folha para rascunho/anotações?
( ) SIM (X) NÃO

Permitido uso de calculadora?
( ) SIM (X) NÃO

Permitido realizar consulta de material?
( ) SIM (X) NÃO

Após a correção da avaliação:

Avaliação passível de revisão?
( ) SIM ( ) NÃO

Q1. *(Perceptron / Regra de aprendizado)*

Um perceptron com ativação degrau (saída 1 se $z \geq 0$; saída 0 se $z < 0$) e taxa de aprendizado $\eta = 0{,}5$ começa uma época com $\mathbf{w} = (1; -1)$ e $b = 0$. Os exemplos são apresentados nesta ordem e a regra de aprendizado do perceptron é aplicada logo depois de cada erro:

exemplo	x_1	x_2	**rótulo** y
1	2	1	1
2	1	2	0
3	0	1	1
4	1	3	0

Ao fim da época, quais são $\mathbf{w}$ e b?

( ) **A)** $\mathbf{w} = (1; -0{,}5)$ e $b = 0{,}5$, com uma única atualização, no exemplo 3.

( ) **B)** $\mathbf{w} = (0{,}5; -2)$ e $b = 0$, com duas atualizações, nos exemplos 3 e 4.

( ) **C)** $\mathbf{w} = (1; -1{,}5)$ e $b = -0{,}5$, com uma única atualização, no exemplo 3.

( ) **D)** $\mathbf{w} = (1{,}5; 1)$ e $b = 1$, com duas atualizações, nos exemplos 3 e 4.

( ) **E)** $\mathbf{w} = (0{,}5; -2)$ e $b = 0{,}5$, com duas atualizações, nos exemplos 3 e 4.

Q2. *(Perceptron multicamadas e história / O problema do XOR)*

Uma rede tem duas entradas binárias, dois neurônios ocultos e uma saída, todos com a ativação degrau do perceptron ($H(z) = 1$ se $z \geq 0$; $H(z) = 0$ se $z < 0$). Os pesos foram escritos à mão:

$$h_1 = H(x_1 - x_2 - 0{,}5), \qquad h_2 = H(x_2 - x_1 - 0{,}5), \qquad y = H(h_1 + h_2 - 0{,}5).$$

x_1	x_2	y
0	0	?
0	1	?
1	0	?
1	1	?

Qual é a coluna y (na ordem da tabela) e o que ela mostra?

( ) **A)** 0, 1, 1, 1: a rede calcula o OU, que um único perceptron já representa; a camada oculta está sobrando.

( ) **B)** 0, 0, 0, 1: a rede calcula o E, a função que Minsky e Papert (1969) mostraram estar fora do alcance do perceptron.

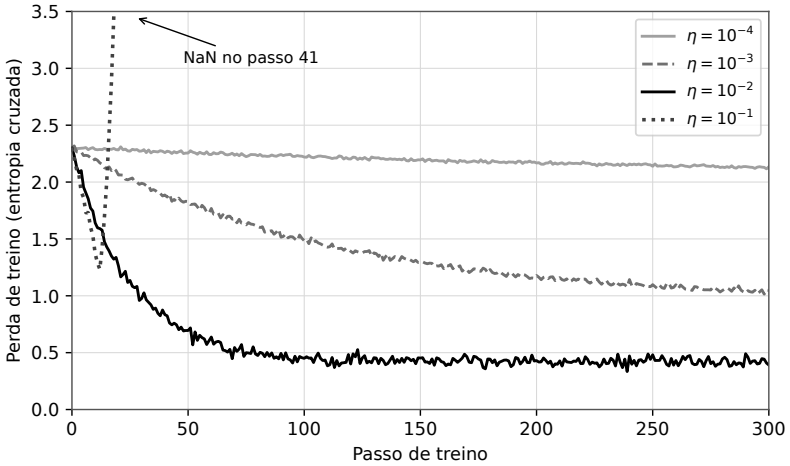
( ) **C)** 0, 1, 1, 0: a rede calcula o XOR, mas só por causa do degrau; com a sigmoide, nenhuma camada oculta o resolveria.

( ) **D)** 0, 1, 1, 0: é o XOR, fora do alcance de um perceptron de uma camada; a retropropagação (1986) permitiu treinar redes como esta.

( ) **E)** 0, 1, 1, 0: a rede calcula o XOR; como ele é linearmente separável em (x_1, x_2), um perceptron sozinho também o aprenderia.

Q3. *(Otimização / Taxa de aprendizado)*

Antes de um treino longo, uma equipe treinou o mesmo modelo (AdamW, 10 classes) por 300 passos com quatro taxas de aprendizado. A figura mostra a perda de treino de cada uma; com $\eta = 10^{-1}$ a perda virou NaN no passo 41.



Qual leitura e qual escolha para o treino longo estão corretas?

( ) **A)** 10^{-1} passou do limite de estabilidade; 10^{-2} é a maior taxa que não explode; no treino longo, usar cerca de $5 \cdot 10^{-3}$.

( ) **B)** 10^{-1} é a melhor, porque é a que mais derruba a perda nos primeiros passos; basta parar o treino antes do NaN.

( ) **C)** 10^{-3}, porque a curva mais suave das quatro indica a taxa que vai convergir melhor num treino longo.

( ) **D)** 10^{-4}, porque uma taxa pequena nunca diverge e, num treino longo, acaba alcançando a perda das outras.

( ) **E)** 10^{-1} com gradient clipping, que impede o NaN e mantém a descida mais rápida que o gráfico mostra.

Q4. *(Regularização / Aumento de dados e validação)*

Um classificador de 10 espécies de aves foi treinado com 6.000 fotos e avaliado numa partição de validação separada. A equipe mediu quatro configurações (acurácia, em %):

	o que muda	**treino**	**validação**
A	nenhuma regularização	100	81
B	A + weight decay e early stopping	97	86
C	B + aumento de dados (recorte aleatório, espelhamento e cor), aplicado ao treino **e à validação**	92	84
D	igual a C, mas a validação é só redimensionada e recortada no centro	92	91

Qual leitura está correta?

( ) **A)** O aumento de dados piorou o modelo: em C, a acurácia de validação caiu de 86% para 84% em relação a B.

( ) **B)** A é a melhor configuração: só ela chegou a 100% no treino, o que mostra que aprendeu todos os padrões.

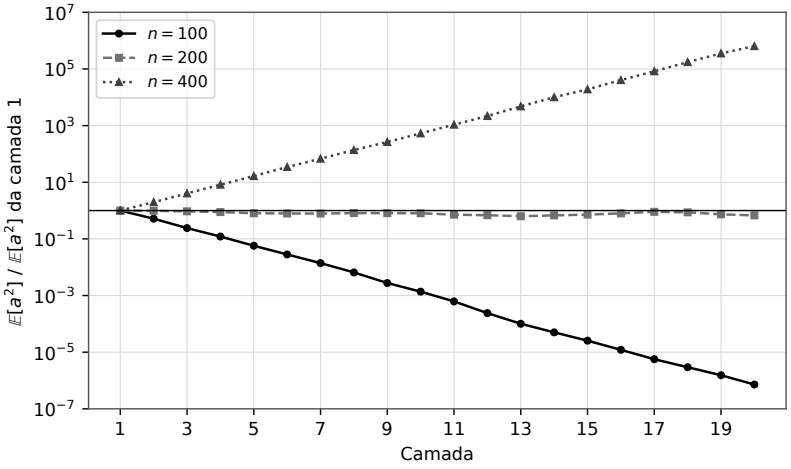
( ) **C)** D é otimista: validar sem aumento esconde os erros em fotos reais, que chegam recortadas e com cores variadas.

( ) **D)** B piorou em relação a A, porque a acurácia de treino caiu de 100% para 97% com a regularização.

( ) **E)** C e D têm o mesmo treino e a diferença vem só da validação aumentada; o aumento de dados ajudou e D é a melhor.

Q5. *(Redes profundas / Inicialização e largura)*

Três MLPs de 20 camadas com ReLU, sem atalhos nem normalização, diferem só na largura n (100, 200 ou 400 unidades por camada). Nos três, cada peso foi sorteado de uma normal de média zero e variância fixa $\sigma^2 = 0{,}01$. O gráfico mostra, na inicialização (antes de qualquer treino), o $\mathbb{E}[a^2]$ de cada camada dividido pelo da camada 1, em escala logarítmica (média geométrica de cinco redes sorteadas por largura).



Por que a mesma inicialização funciona para uma largura e falha para as outras? O que corrige isso?

( ) **A)** A largura muda a profundidade efetiva: redes largas têm mais unidades ativas por camada e por isso o sinal cresce na de 400.

( ) **B)** A ReLU satura nas redes largas: somar mais entradas leva as pré-ativações para a região plana da função.

( ) **C)** O fator por camada é $n\sigma^2 \cdot \frac{1}{2}$: 0,5, 1 e 2 para n = 100, 200 e 400; com $\sigma^2 = 2/n$ (He), ele vale 1 nas três.

( ) **D)** O fator por camada é $n\sigma^2$: 1, 2 e 4; só a rede de 100 deveria ser estável e a figura indica um erro de medida.

( ) **E)** É a taxa de aprendizado: redes mais largas pedem η menor e a de 400 explode por isso já na inicialização.

Q6. *(Redes convolucionais / Formatos e parâmetros)*

Uma CNN recebe imagens RGB 64×64 e termina num `Linear` de 10 classes. O aluno precisa completar a tabela para escrever a entrada desse `Linear`:

camada	**formato da saída**
entrada	$3 \times 64 \times 64$
`Conv2d(3, 16, 3, padding=1)` + ReLU	$16 \times 64 \times 64$
`MaxPool2d(2)`	$16 \times 32 \times 32$
`Conv2d(16, 32, 3, stride=2, padding=1)` + ReLU	?
`MaxPool2d(2)`	?
`Flatten`	?
`Linear(?, 10)`	10

Qual é o formato antes do `Flatten` e quantos parâmetros (com viés) tem o `Linear`? Quantos ele teria com `AdaptiveAvgPool2d(1)` (pooling global) logo antes do `Flatten`?

( ) **A)** $32 \times 16 \times 16$; `Linear(8192, 10)`, com 81.930 parâmetros; com o pooling global, 330.

( ) **B)** $32 \times 8 \times 8$; `Linear(2048, 10)`, com 20.490 parâmetros; com o pooling global, 330.

( ) **C)** $32 \times 7 \times 7$; `Linear(1568, 10)`, com 15.690 parâmetros; com o pooling global, 330.

( ) **D)** $32 \times 8 \times 8$; `Linear(2048, 10)`, com 20.480 parâmetros; com o pooling global, 20.

( ) **E)** $32 \times 8 \times 8$; `Linear(2048, 10)`, com 20.490 parâmetros; o pooling global não muda essa conta.

Q7. *(Vision Transformers e transfer learning / LoRA)*

Uma equipe adapta um ViT-B/16 pré-treinado (cerca de 86,6 milhões de parâmetros, $d = 768$, 12 blocos) com LoRA de posto $r = 4$ nas quatro projeções de atenção de cada bloco (W_q, W_k, W_v e W_o, todas 768×768); o resto fica congelado. Sem contar a cabeça nova, quantos parâmetros são treinados? Com o AdamW guardando 12 bytes por parâmetro treinável (cópia mestra em fp32 e dois momentos), quanto ocupa o estado do otimizador?

( ) **A)** 147.456, pois cada matriz ganha $r \cdot d$ números; o estado do AdamW fica em cerca de 1,8 MB.

( ) **B)** 28.311.552, pois cada $\Delta W = BA$ tem 768×768 entradas; o estado do AdamW fica em cerca de 340 MB.

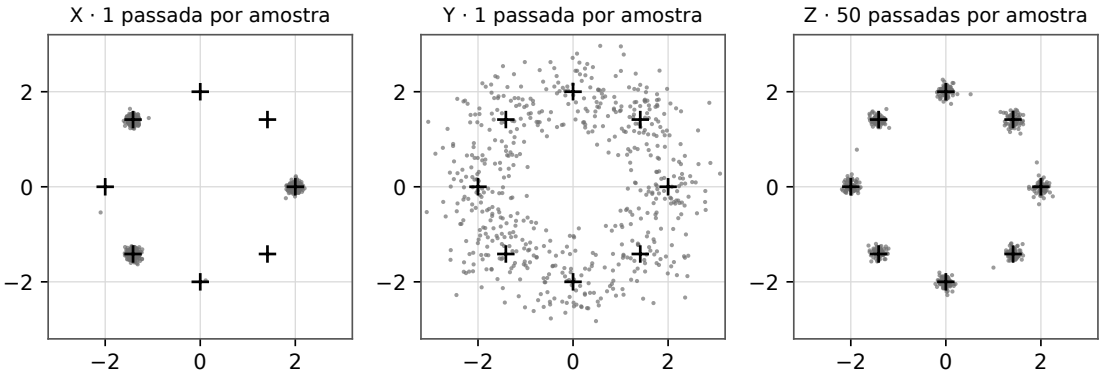
( ) **C)** 294.912; o estado do AdamW não muda, porque o otimizador guarda os momentos de todos os pesos do modelo.

( ) **D)** 294.912; o estado do AdamW fica em cerca de 0,6 MB, porque os momentos usam 2 bytes por parâmetro.

( ) **E)** 294.912, cerca de 0,3% do modelo; o estado do AdamW cai de cerca de 1 GB (fine-tuning completo) para cerca de 3,5 MB.

Q8. *(Modelos generativos / O que as amostras revelam)*

Três modelos foram treinados nos mesmos dados, oito aglomerados dispostos num anel (as cruzes marcam os centros): uma GAN, um VAE e um modelo de flow matching, não necessariamente nessa ordem. A figura mostra 600 amostras de cada um e quantas passadas pela rede cada amostra custou.



Qual leitura está correta?

( ) **A)** X é a GAN, que escolhe modos como a KL reversa; Y é o VAE, que cobre todos (inclusive o vale entre eles) como a KL para frente.

( ) **B)** X é o VAE: o borrão do VAE concentra as amostras em poucos modos, enquanto a GAN, Y, espalha as amostras pelo anel.

( ) **C)** Y é a GAN com colapso de modos: as amostras entre os aglomerados mostram que ela se fixou em poucos deles.

( ) **D)** Z é o VAE, porque só um modelo treinado por verossimilhança cobre os oito modos com amostras tão nítidas.

( ) **E)** X e Y diferem só por treino curto: com mais épocas, X cobriria os oito modos e Y pararia de pôr amostras entre eles.

Q9. *(Autoencoders variacionais / A KL de cada dimensão)*

Um VAE com latente de 8 dimensões foi treinado em fotos de roupas com $\beta = 1$. Na validação, a equipe registrou a KL média até o prior $\mathcal{N}(0, I)$, separada por dimensão do latente, em nats:

dimensão	1	2	3	4	5	6	7	8
KL média	3,1	2,4	0,01	1,7	0,02	0,00	2,9	0,01

Qual leitura está correta?

( ) **A)** As dimensões 3, 5, 6 e 8 são as mais informativas, porque o encoder as usa sem pagar quase nada de KL.

( ) **B)** A KL total, de cerca de 10 nats, indica colapso do posterior: o encoder ignora a entrada em todas as dimensões.

( ) **C)** Nas dimensões com KL perto de zero, $\sigma \to 0$: o encoder as codifica sem incerteza nenhuma.

( ) **D)** As dimensões 3, 5, 6 e 8 estão inativas: nelas $q(z \mid x) \approx \mathcal{N}(0, 1)$ para toda imagem e o decoder não tem o que ler.

( ) **E)** Aumentar o β reativaria as dimensões 3, 5, 6 e 8, porque o termo KL passaria a pesar mais nelas.

Q10. *(GANs / A perda do gerador)*

No laço de treino de uma GAN, `D` devolve logits e `bce(logits, alvo)` é a entropia cruzada binária com sigmoide (a `BCEWithLogitsLoss`). O passo do gerador está assim:

```
fake = G(z)
loss_G = -bce(D(fake), torch.zeros(B))      # B: tamanho do lote
opt_G.zero_grad(); loss_G.backward(); opt_G.step()
```

Nos primeiros 2.000 passos, $D(G(z))$ fica em torno de 0,02 e as amostras quase não mudam. O que explica isso e o que mudar?

( ) **A)** Nada está errado: minimizar $-$`bce(D(fake), 0)` é o objetivo do gerador no artigo original; falta só treinar mais.

( ) **B)** Falta `fake.detach()`: sem ele, o gradiente do gerador vaza para o discriminador, que por isso vence cedo.

( ) **C)** É a perda minimax, $\log(1 - D(G(z)))$: com $D(G(z)) \approx 0{,}02$, o gradiente no logit é só $-0{,}02$; `bce(D(fake), ones)` dá $-0{,}98$.

( ) **D)** O sinal está trocado: o gerador está maximizando $\log(1 - D(G(z)))$; basta tirar o sinal de menos da perda.

( ) **E)** O jogo chegou ao equilíbrio, em que o discriminador não distingue reais de falsos; não há nada a mudar.

Q11. *(CLIP / Classificação zero-shot)*

Um parque usa o CLIP para classificar fotos de armadilhas fotográficas, com um prompt por classe e temperatura $\tau = 0{,}01$ (logit = 100 $\times$ cosseno). Para uma foto, os cossenos com os prompts são:

classe	lobo-guará	cachorro-do-mato	jaguatirica
cosseno	0,29	0,27	0,22

Use $e^{-2} \approx 0{,}14$, $e^{-7} \approx 0{,}001$ e $e^{-9} \approx 0{,}0001$. Qual é a classe prevista e com que probabilidade? O que acontece se o parque acrescentar a classe "tamanduá-bandeira", cujo cosseno com a foto é 0,20?

( ) **A)** Lobo-guará, com cerca de 0,88; a nova classe quase não mexe nesse valor e não exige treinar nada.

( ) **B)** Lobo-guará, com 0,29, o próprio cosseno; a nova classe não muda esse valor, que não depende das outras.

( ) **C)** Lobo-guará, com cerca de 0,37 (0,29 sobre a soma dos cossenos); a nova classe o baixa para cerca de 0,30.

( ) **D)** Lobo-guará, com cerca de 0,88; a nova classe exige retreinar a camada linear do classificador com fotos dela.

( ) **E)** Nenhuma: cossenos abaixo de 0,3 mostram que a foto não casa com nenhum prompt; a resposta deve ser descartada.

Q12. *(Stable Diffusion / Classifier-free guidance)*

Num passo de amostragem do Stable Diffusion com escala de guidance $w = 7{,}5$, a U-Net devolveu, para duas componentes do latente, as previsões de ruído abaixo:

componente	**sem o prompt**, $\epsilon_\theta(z_t, \varnothing)$	**com o prompt**, $\epsilon_\theta(z_t, c)$
1	0,10	0,30
2	$-0{,}20$	$-0{,}20$

Qual é a previsão guiada $\tilde{\epsilon}$ dessas duas componentes?

( ) **A)** $(0{,}30;\ -0{,}20)$: com o prompt presente, a previsão guiada é a própria previsão condicional.

( ) **B)** $(2{,}25;\ -1{,}50)$: o guidance multiplica a previsão condicional pela escala w, componente a componente.

( ) **C)** $(1{,}60;\ -0{,}20)$: a componente 2 não muda, porque nela o prompt não altera a previsão de ruído.

( ) **D)** $(1{,}80;\ -0{,}20)$: o guidance parte da previsão condicional e soma w vezes a diferença entre as duas.

( ) **E)** $(1{,}60;\ -1{,}50)$: a escala w amplifica as duas componentes, inclusive a que não depende do prompt.

Q13. *(Flow matching / O interpolante e a convenção do tempo)*

Um passo de treino de flow matching com o interpolante linear sorteia o ruído $x_0 = (1;\ -2)$, o dado $x_1 = (3;\ 2)$ e $t = 0{,}25$. Na convenção da aula, $t = 0$ é o ruído e $t = 1$ são os dados. O código de um colega segue a convenção do artigo do SD3 (dados em $t = 0$, ruído em $t = 1$) e sorteou os mesmos dois pontos e o mesmo t. Que entrada x_t e que alvo de regressão cada versão usa?

( ) **A)** Aula: $x_t = (2{,}5;\ 1)$ e alvo $(2;\ 4)$. Colega: $x_t = (1{,}5;\ -1)$ e alvo $(-2;\ -4)$.

( ) **B)** Aula: $x_t = (1{,}5;\ -1)$ e alvo $(2;\ 4)$. Colega: $x_t = (2{,}5;\ 1)$ e alvo $(-2;\ -4)$.

( ) **C)** Aula: $x_t = (1{,}5;\ -1)$ e alvo $(3;\ 2)$, o dado. Colega: o mesmo x_t e alvo $(1;\ -2)$, o ruído.

( ) **D)** As duas: $x_t = (1{,}5;\ -1)$ e alvo $(2;\ 4)$, porque o interpolante linear é simétrico no tempo.

( ) **E)** Aula: $x_t = (1{,}5;\ -1)$ e alvo $(2;\ 4)$. Colega: $x_t = (2{,}5;\ 1)$ e o mesmo alvo, $(2;\ 4)$.

Q14. *(Diffusion Transformers e geração autorregressiva / Escolha do gerador)*

Uma empresa vai lançar dois produtos de geração de imagens e precisa escolher o desenho do gerador de cada um:

produto	o que ele precisa fazer
1	API de texto para imagem em 1024×1024, avaliada pela qualidade por GPU-hora
2	assistente que conversa, lê fotos do usuário, raciocina em texto e devolve a foto editada na mesma conversa

Qual escolha está mais bem justificada?

( ) **A)** 1: autorregressivo em ordem raster, pela verossimilhança exata; 2: DiT com cross-attention, que já lê o texto do usuário.

( ) **B)** Os dois com MMDiT: na atenção conjunta, a imagem reescreve os tokens de texto, então o mesmo modelo já responde em texto.

( ) **C)** Os dois autorregressivos sobre tokens VQ, porque um codebook grande dá mais fidelidade que qualquer VAE contínuo.

( ) **D)** 1: um transformer de difusão ou de flow; 2: um modelo único sobre uma sequência que intercala texto e imagem.

( ) **E)** 1: MaskGIT, porque oito rodadas paralelas superam qualquer DiT em qualidade; 2: VAR, que gera do grosso ao fino.

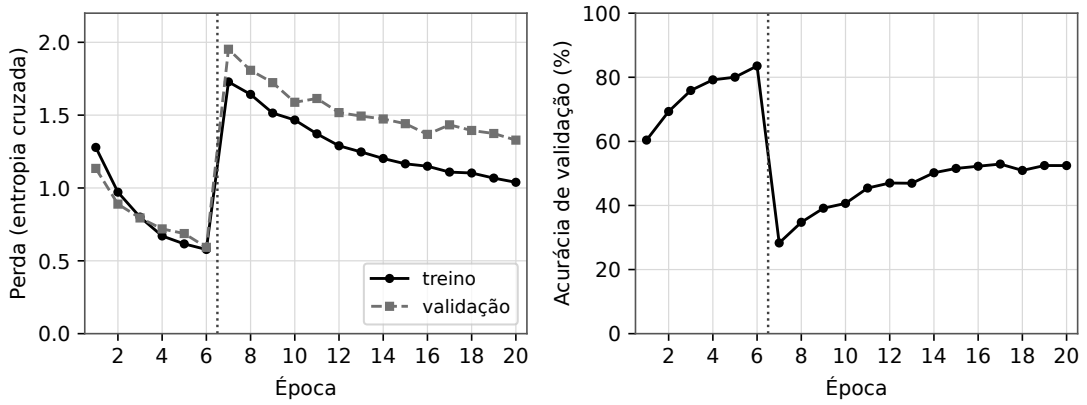
Q15. *(Dissertativa – Um projeto de deep learning de ponta a ponta – 3,0 pontos)*

Uma cooperativa de cafeicultores quer um app que classifique a foto de uma folha de café, tirada no celular, em uma de cinco classes: *sadia*, *ferrugem*, *cercosporiose*, *bicho-mineiro* ou *phoma*. Agrônomos rotularam 2.400 fotos (480 por classe), já divididas em treino (1.920) e validação (480); um conjunto de teste será coletado na próxima safra. Ferrugem e cercosporiose se distinguem sobretudo pela cor e pelo formato das manchas. A equipe tem dois backbones pré-treinados no ImageNet: uma ResNet-50 (CNN, 25 milhões de parâmetros) e um ViT-B/16 (86 milhões).

(a) Modelo e transferência (1,0). Que tipo de problema de aprendizado é este? Qual backbone você usaria e por quê? O que você congelaria ou adaptaria (linear probe, fine-tuning, LoRA) e em que ordem? Diga a entrada, a saída e a perda.

(b) Treino (1,0). Escreva, em Python/PyTorch (pode ser esquemático), o laço de treino com o otimizador, o agendamento da taxa de aprendizado, a regularização escolhida (justifique o aumento de dados para este domínio), a validação em `model.eval()` e o early stopping.

(c) Diagnóstico (1,0). Outra equipe treinou só a cabeça nova por 6 épocas (AdamW, $\eta = 10^{-2}$). Na época 7 ela descongelou o backbone inteiro (sem mudar mais nada) e obteve as curvas abaixo ($\ln 5 \approx 1{,}61$). O que a figura mostra? O que você mudaria primeiro e por quê?



NOME: _____

	A	B	C	D	E
Q1					
Q2					
Q3					
Q4					
Q5					
Q6					
Q7					

	A	B	C	D	E
Q8					
Q9					
Q10					
Q11					
Q12					
Q13					
Q14					

Q15 (a) modelo e transferência:

NOME: _____

Q15 (b) treino (pode ser código esquemático):

Q15 (c) diagnóstico (se necessário, use o verso):