

Instruções: Cada questão possui exatamente uma alternativa correta.

Questões

- I) Um escalonador *cosine decay* sem *warm-up* é configurado com $\eta_{\max} = 10^{-3}$, $\eta_{\min} = 10^{-4}$ e duração total $T = 10\,000$ passos, seguindo a fórmula vista em aula:

$$\eta_t = \eta_{\min} + \frac{\eta_{\max} - \eta_{\min}}{2} \left(1 + \cos \frac{\pi t}{T} \right).$$

Qual o valor de η_t na metade do treinamento ($t = 5\,000$)?

- a) 5.5×10^{-4} .
- b) 1.0×10^{-3} .
- c) 4.5×10^{-4} .
- d) 1.0×10^{-4} .

- II) Um modelo possui $N = 2 \times 10^9$ parâmetros (2B). Um *fine-tuning* completo com AdamW em *mixed precision* mantém simultaneamente em VRAM:

- pesos em fp16 (2 B/par)
- gradientes em fp16 (2 B/par)
- cópia mestre dos pesos em fp32 (4 B/par)
- primeiro momento Adam m em fp32 (4 B/par)
- segundo momento Adam v em fp32 (4 B/par)

Qual o total de VRAM requerido por esses componentes (sem contar ativações)?

- a) 32 GB.
- b) 4 GB.
- c) 8 GB.
- d) 24 GB.

- III) A perda do SFT é

$$\mathcal{L}_{\text{SFT}}(\theta) = - \sum_{t \in \mathcal{R}} \log P_{\theta}(x_t \mid x_{<t}),$$

ou seja, somente os tokens da resposta ($\mathcal{R}$) contribuem. Um estudante sugere calcular a perda em todos os tokens (prompt + resposta), argumentando que “mais supervisão = melhor aprendizado”. Qual a principal falha desse raciocínio?

- a) O gradiente passa a incluir contribuições dos tokens de instrução, que seguem formatos repetitivos em todos os exemplos; o modelo gastaria capacidade aprendendo a *gerar instruções* em vez de aprender a *responder* a elas.
- b) A perda ficaria numericamente indefinida porque os tokens do prompt não pertencem ao vocabulário do modelo.

- c) O *teacher forcing* falharia, pois tokens especiais como `<|im_start|>` não podem ser previstos.
- d) A normalização por $|\mathcal{R}|$ resultaria em divisão por zero, já que todos os tokens passariam a contar.
- IV) Uma equipe treina um modelo de 7B em fp16 *mixed precision* e observa NaN frequentes nos gradientes. Ao trocar para bf16, o problema desaparece sem precisar de *loss scaling*. Qual a explicação correta para esse comportamento?
- a) BF16 tem 8 bits de expoente (mesma faixa dinâmica do fp32: $\sim 10^{-38}$ a $\sim 10^{38}$), enquanto fp16 tem 5 bits de expoente e satura próximo de 65 504. Gradientes de redes profundas extrapolam esse limite, causando Inf/NaN; a faixa larga do bf16 elimina o *overflow* sem necessidade de escalonamento.
- b) BF16 tem 16 bits de mantissa, oferecendo mais precisão que os 10 bits do fp16, eliminando os erros de arredondamento que causavam os NaN.
- c) BF16 é *lossless* para pesos de redes neurais, ao contrário do fp16, que sempre introduz erros de quantização.
- d) BF16 usa o dobro de bits do fp16 (32 vs. 16), o que explica o ganho de estabilidade numérica.
- V) O *Reward Model* (RM) é treinado com a perda de Bradley-Terry:

$$\mathcal{L}_{\text{RM}} = -\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l)).$$

Para um par de treino, o RM atribui $r_\phi(x, y_w) = 1.8$ e $r_\phi(x, y_l) = 0.3$. Usando $\sigma(1.5) \approx 0.818$, qual o valor de $\mathcal{L}_{\text{RM}}$?

- a) ≈ 0.201 .
- b) ≈ 1.702 .
- c) ≈ 0.693 .
- d) ≈ 1.5 .
- VI) Durante o alinhamento de um modelo de 7B, o RLHF-PPO mantém **4 modelos** simultâneos em VRAM (política, referência SFT, *reward model*, *value function*); o DPO mantém apenas **2** (política + referência congelada). Suponha cada cópia bf16 ≈ 14 GB e que o estado do otimizador AdamW apenas da *política* ocupa ≈ 100 GB (ignorando ativações). Quanto cada método consome e qual a economia do DPO?
- a) PPO ≈ 156 GB, DPO ≈ 128 GB (economia ≈ 28 GB).
- b) PPO ≈ 114 GB, DPO ≈ 114 GB (economia ≈ 0).
- c) PPO ≈ 56 GB, DPO ≈ 28 GB (economia ≈ 28 GB).
- d) PPO ≈ 28 GB, DPO ≈ 14 GB (economia ≈ 14 GB).
- VII) Um *transformer* com $d_{\text{model}} = 2\,048$ possui quatro projeções de atenção por camada (Q, K, V, O), cada uma de formato $2\,048 \times 2\,048$. Aplicando LoRA com *rank* $r = 16$ nas quatro projeções ($h = W_0x + \frac{\alpha}{r} B A x$, com $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$), quantos parâmetros treináveis são adicionados por camada e que fração representam dos pesos originais de atenção?
- a) 262 144 params ($\approx 1.56\%$ dos 16.78 M originais).

- b) 131 072 params ($\approx 0.78\%$).
- c) 524 288 params ($\approx 3.12\%$).
- d) 65 536 params ($\approx 0.39\%$).

VIII) No LoRA, A é inicializado com distribuição normal gaussiana e B com **zeros**. Um estudante pergunta: “por que não inicializar *ambos* aleatoriamente?”. Qual a justificativa correta para a escolha $B = 0$?

- a) Com $B = 0$, temos $\Delta W = \frac{\alpha}{r}BA = 0$ no passo inicial, logo o modelo adaptado começa *exatamente* do pré-treinamento. Se B fosse aleatório, $\Delta W \neq 0$ adicionaria ruído às projeções pré-treinadas, degradando o ponto de partida.
- b) Inicializar B aleatoriamente faria com que o adaptador LoRA competisse com os pesos originais, reduzindo o número de parâmetros treináveis efetivos.
- c) $B = 0$ é matematicamente necessário para que a decomposição $\Delta W = BA$ tenha *rank* exatamente r : inicialização aleatória quebraria a restrição de baixo *rank*.
- d) Com B aleatório, o fator α/r causaria explosão de gradiente nas primeiras iterações.

IX) Um bloco do Flash Attention processa os escores $s = [1, 2, 0, 3]$ em dois tiles: $B_1 = [1, 2]$ e $B_2 = [0, 3]$. A recursão do *online softmax* é

$$m^{\text{new}} = \max(m^{\text{old}}, \max_j s_j), \quad \ell^{\text{new}} = e^{m^{\text{old}} - m^{\text{new}}} \ell^{\text{old}} + \sum_j e^{s_j - m^{\text{new}}}.$$

Qual o valor de ℓ após processar os dois tiles? (Use $e^{-1} \approx 0.368$, $e^{-2} \approx 0.135$, $e^{-3} \approx 0.050$.)

- a) ≈ 1.553 .
- b) ≈ 2.418 .
- c) ≈ 1.368 .
- d) ≈ 1.050 .

X) Uma equipe faz SFT de um modelo usando $\eta = 3 \times 10^{-4}$ (mesma LR usada no pré-treino) por 10 épocas. A perda SFT no conjunto de treino cai normalmente, mas *benchmarks* gerais (MMLU, HellaSwag) **degradam visivelmente** ao longo das épocas. Qual diagnóstico e correção são adequados?

- a) *Catastrophic forgetting*: LR alto por muitas épocas sobrescreve representações adquiridas no pré-treinamento, mesmo com a perda SFT em queda. Correção: reduzir LR para 10^{-6} – 2×10^{-5} e limitar a 1–3 épocas.
- b) Comportamento esperado: SFT sempre descarta conhecimento do pré-treinamento; a correção é treinar do zero com os dados SFT.
- c) *Overfitting* nos próprios *benchmarks*; a correção é adicionar mais *benchmarks* ao conjunto de avaliação.
- d) Explosão de gradiente causada pelo LR alto; a correção é aumentar o *weight decay* para 1.0.

- XI) 🦴 Em uma iteração de GRPO para um problema com *reward* binário (1 = correto, 0 = errado), um prompt produz um grupo de $G = 6$ respostas com recompensas $r = [1, 1, 0, 1, 0, 1]$. A vantagem é

$$\hat{A}_i = \frac{r_i - \bar{r}}{\sigma_r},$$

onde $\bar{r}$ e σ_r são a média e o desvio-padrão do grupo. Quais os valores de $\hat{A}$ para as respostas corretas e incorretas?

- a) $\hat{A}_{\text{correto}} \approx +0.707$; $\hat{A}_{\text{errado}} \approx -1.414$.
- b) $\hat{A}_{\text{correto}} = +1.000$; $\hat{A}_{\text{errado}} = -1.000$.
- c) $\hat{A}_{\text{correto}} = +1/3$; $\hat{A}_{\text{errado}} = -2/3$.
- d) $\hat{A}_{\text{correto}} = +0.500$; $\hat{A}_{\text{errado}} = 0$.

- XII) 🦴 A perda do DPO é

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right).$$

Em um exemplo com $\beta = 0.1$, $\log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} = 2$ e $\log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} = -1$, qual o valor de $\mathcal{L}_{\text{DPO}}$? (Use $\sigma(0.3) \approx 0.574$, $\sigma(3) \approx 0.953$, $\sigma(-0.3) \approx 0.426$.)

- a) ≈ 0.555 .
- b) ≈ 0.048 .
- c) ≈ 0.853 .
- d) ≈ 0.300 .

Soluções

I — Resposta: (a)

Em $t = T/2$: $\cos(\pi \cdot 0.5) = \cos(\pi/2) = 0$, logo $1 + \cos(\pi/2) = 1$. Portanto,

$$\eta_{T/2} = 10^{-4} + \frac{10^{-3} - 10^{-4}}{2} \cdot 1 = 10^{-4} + 4.5 \times 10^{-4} = 5.5 \times 10^{-4}.$$

A opção (b) ignora totalmente o termo com cosseno. A opção (c) omite a parcela $\eta_{\min}$ da fórmula. A opção (d) confunde $\cos(\pi/2) = 0$ com $\cos(\pi) = -1$, obtendo $\eta_{\min}$, valor correto apenas no fim do treino ($t = T$).

II — Resposta: (a)

Somando os componentes em bytes por parâmetro: $2+2+4+4+4 = 16$ B/par. Aplicando a $N = 2 \times 10^9$: $2 \times 10^9 \times 16 = 3.2 \times 10^{10}$ bytes = 32 GB. As opções (b), (c) e (d) ignoram subconjuntos dos componentes: somente pesos, pesos+gradientes e um dos momentos do Adam, respectivamente.

III — Resposta: (a)

Os tokens de instrução seguem padrões repetitivos (`<|im_start|>user`, “Explique...”, etc.). Se forem incluídos na perda, o gradiente é dominado por eles e o modelo gasta capacidade modelando a *distribuição das instruções* em vez de aprender a gerar boas respostas. Mascaram o prompt garante que o aprendizado se concentre em $P(\text{resposta} \mid \text{instrução})$, não em $P(\text{instrução})$. A opção (b) é falsa: tokens de prompt estão no vocabulário (caso contrário, não poderiam sequer ser alimentados como entrada). A opção (c) confunde mascaramento de *loss* com ausência do token no vocabulário. A opção (d) é falsa: a normalização é por $|\mathcal{R}|$ (tokens da resposta), e incluir os tokens do prompt apenas aumentaria $|\mathcal{R}|$, não o zeraria.

IV — Resposta: (a)

A diferença essencial entre fp16 e bf16 é o número de bits de expoente: fp16 usa 5 (limite $\approx 65\,504$) e bf16 usa 8 (mesmo limite do fp32, $\approx 3.4 \times 10^{38}$). Gradientes grandes provocam *overflow* em fp16, gerando Inf/NaN; a faixa larga do bf16 elimina esse problema sem exigir *loss scaling*. A opção (b) inverte o trade-off: bf16 tem *menos* mantissa (7 bits) que o fp16 (10). O ganho não é precisão, é faixa. A opção (c) é falsa: ambos são lossy. A opção (d) é falsa: ambos ocupam 16 bits.

V — Resposta: (a)

$\Delta r = r_\phi(x, y_w) - r_\phi(x, y_l) = 1.8 - 0.3 = 1.5$. $\sigma(1.5) \approx 0.818$, portanto $\mathcal{L}_{\text{RM}} = -\ln(0.818) \approx 0.201$. A opção (b) inverte o sinal da diferença, aplicando $\sigma(-1.5) \approx 0.182$ ($-\ln(0.182) \approx 1.702$): seria a perda se o RM ordenasse *incorretamente*. A opção (c) usa $\sigma(0) = 0.5$, equivalente a ignorar totalmente Δr . A opção (d) retorna o *logit* sem passá-lo pelo σ nem aplicar $-\ln$.

VI — Resposta: (a)

PPO: 4 cópias em bf16 + estado do otimizador da política = $4 \cdot 14 + 100 = 56 + 100 = 156$ GB.

DPO: 2 cópias em bf16 + estado do otimizador da política = $2 \cdot 14 + 100 = 28 + 100 = 128$ GB.

Economia: $156 - 128 = 28$ GB, exatamente o custo das duas cópias que o DPO dispensa (*reward model* e *value function*). A opção (b) é falsa: o PPO *inclui* duas cópias adicionais frozen, portanto sua memória é maior. A opção (c) ignora o estado do otimizador da política (100 GB). A opção (d) reduz incorretamente cada método a uma única cópia.

VII — Resposta: (a)

Cada projeção LoRA adiciona $|B| + |A| = d \cdot r + r \cdot k = 2 \cdot d \cdot r$ parâmetros (como $d = k = 2\,048$). Para quatro projeções:

$$4 \times 2 \cdot 2\,048 \cdot 16 = 4 \times 65\,536 = 262\,144 \text{ params.}$$

Pesos originais: $4 \times 2\,048^2 = 16\,777\,216 \approx 16.78$ M. Fração: $262\,144 / 16\,777\,216 \approx 1.56\%$. A opção (b) conta apenas $d \cdot r$ por projeção, esquecendo a segunda matriz. A opção (c) duplica erroneamente o resultado. A opção (d) considera apenas uma das quatro projeções.

VIII — Resposta: (a)

O ponto de partida do LoRA é o modelo pré-treinado intacto: exige-se $\Delta W = \frac{\alpha}{r} BA = 0$ na inicialização. Zerar B (com A aleatório) satisfaz essa condição. Se *ambos* fossem aleatórios, $\Delta W \neq 0$ adicionaria ruído aos pesos congelados, destruindo o ponto de partida calibrado do pré-treinamento. O modelo teria de “reaprender” antes mesmo de começar a se adaptar aos dados de ajuste. A opção (b) não faz sentido dimensional: inicialização não altera contagem de parâmetros treináveis. A opção (c) é falsa: o *rank* de BA é determinado pela forma das matrizes, não pelos valores iniciais. A opção (d) confunde o papel de α/r (fator de escala constante, não amplificador de gradiente).

IX — Resposta: (a)

Processando $B_1 = [1, 2]$:

$$\begin{aligned} m^{(1)} &= \max(1, 2) = 2 \\ \ell^{(1)} &= e^{1-2} + e^{2-2} = 0.368 + 1 = 1.368. \end{aligned}$$

Processando $B_2 = [0, 3]$:

$$\begin{aligned} m^{(2)} &= \max(2, 3) = 3 \\ \ell^{(2)} &= e^{m^{(1)} - m^{(2)}} \cdot \ell^{(1)} + e^{0-3} + e^{3-3} \\ &= e^{-1} \cdot 1.368 + e^{-3} + e^0 \\ &\approx 0.368 \cdot 1.368 + 0.050 + 1 \\ &\approx 0.503 + 0.050 + 1 = \mathbf{1.553}. \end{aligned}$$

Conferência (softmax global): $e^{1-3} + e^{2-3} + e^{0-3} + e^{3-3} = 0.135 + 0.368 + 0.050 + 1 = 1.553$. ✓ A opção (b) omite o fator de rescala $e^{m^{\text{old}} - m^{\text{new}}}$, tratando $\ell^{(1)}$ como se já estivesse normalizado para $m^{(2)}$. A opção (c) ignora B_2 . A opção (d) ignora o histórico $\ell^{(1)}$.

X — Resposta: (a)

O padrão é clássico de *catastrophic forgetting*: a perda SFT decresce porque o modelo aprende o *formato* dos dados de ajuste, mas as muitas atualizações com LR alto modificam significativamente os pesos, apagando representações úteis para tarefas gerais. O remédio canônico é LR muito menor que no pré-treinamento (10^{-6} – 2×10^{-5}) e apenas 1–3 épocas. A opção (b) contradiz a hipótese do alinhamento superficial (LIMA): o SFT ensina formato e estilo, não substitui o pré-treinamento. A opção (c) é implausível: *benchmarks* não estão no conjunto de treino, então *overfitting* neles não faz sentido. A opção (d) descreve um sintoma diferente: explosão de gradiente geraria NaN agudos, não degradação gradual. Além disso, aumentar *weight decay* não corrige explosões e pioraria o *forgetting*.

XI — Resposta: (a)

Com rewards $[1, 1, 0, 1, 0, 1]$ e $G = 6$:

$$\begin{aligned}\bar{r} &= \frac{4}{6} = \frac{2}{3} \approx 0.667, \\ \sigma_r^2 &= \mathbb{E}[r^2] - \bar{r}^2 = \frac{4}{6} - \frac{4}{9} = \frac{6}{9} - \frac{4}{9} = \frac{2}{9}, \\ \sigma_r &= \frac{\sqrt{2}}{3} \approx 0.471.\end{aligned}$$

Portanto,

$$\begin{aligned}\hat{A}_{\text{correto}} &= \frac{1 - 2/3}{\sqrt{2}/3} = \frac{1/3}{\sqrt{2}/3} = \frac{1}{\sqrt{2}} \approx +0.707, \\ \hat{A}_{\text{errado}} &= \frac{0 - 2/3}{\sqrt{2}/3} = -\frac{2}{\sqrt{2}} = -\sqrt{2} \approx -1.414.\end{aligned}$$

A magnitude maior para as respostas erradas (minoritárias) é esperada: o z-score pesa mais a cauda menos populosa. A opção (b) usa os valores do slide, que valem apenas quando *metade* do grupo acerta ($G = 4$, 2 corretos, 2 errados), situação em que $\sigma_r = 0.5$. A opção (c) esquece a normalização por σ_r . A opção (d) substitui a *baseline* correta (média) por $\max(r) = 1$.

XII — Resposta: (a)

Argumento do σ :

$$\beta \left(\log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) = 0.1 \cdot (2 - (-1)) = 0.3.$$

$\sigma(0.3) \approx 0.574$, portanto $\mathcal{L}_{\text{DPO}} = -\ln(0.574) \approx 0.555$. A opção (b) esquece o β , aplicando $\sigma(3) \approx 0.953$, subestimando dramaticamente a perda. A opção (c) inverte o sinal, usando $\sigma(-0.3)$ (corresponderia a uma *preferência errada*: y_l mais provável que y_w). A opção (d) retorna o argumento do σ sem aplicar $-\log \sigma$.