Reinforcement Learning of Large Language Models
E. Ryu
Spring 2025

UCLA

Homework exercises

Problem 1: *Monotonicity of Bellman operators.* Let $\gamma \in (0,1)$, $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$, and $|r| \leq R < \infty$ almost surely. Let π be a policy, not necessarily optimal. Let $\mathcal{B}^\pi$ be the Bellman operator for π and $\mathcal{B}^\star$ the Bellman optimality operator. Show that for any $V : \mathcal{S} \to \mathbb{R}$,

$$\mathcal{B}^\pi[V] \leq \mathcal{B}^\star[V].$$

Also show that for any $U : \mathcal{S} \to \mathbb{R}$ and $V : \mathcal{S} \to \mathbb{R}$ such that $U \leq V$,

$$\mathcal{B}^\star[U] \leq \mathcal{B}^\star[V].$$

Problem 2: *Bellman operators for Q are contractions.* Let $\gamma \in (0,1)$, $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$, and $|r| \leq R < \infty$ almost surely. Show that the $\mathcal{B}^\pi$ and $\mathcal{B}^\star$ for Q are γ-contractions.

Problem 3: *Optimal Q-function dominates all Q-functions.* Let $\gamma \in (0,1)$, $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$, and $|r| \leq R < \infty$ almost surely. Let $\pi^\star$ be an optimal policy, i.e., assume

$$V^{\pi^\star}(s) \geq V^\pi(s), \qquad \forall\, s \in \mathcal{S}, \text{ policy } \pi.$$

Show that

$$Q^{\pi^\star}(s,a) \geq Q^\pi(s,a), \qquad \forall\, s \in \mathcal{S},\ a \in \mathcal{A}, \text{ policy } \pi.$$

Problem 4: *Optimal policies form a convex set.* Let $\gamma \in (0,1)$, $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$, and $|r| \leq R < \infty$ almost surely. Show the following:

(a) Show that a policy π (not necessarily deterministic) is optimal if and only if

$$V^\pi(s) = V^\star(s), \qquad \forall\, s \in \mathcal{S}.$$

(b) Show that

$$\mathop{\mathbb{E}}_{(r,s')\sim p(\cdot,\cdot\,|\,s,a)}[r + \gamma V^\star(s') \,|\, s,a] = Q^\star(s,a), \qquad \forall\, s \in \mathcal{S},\ a \in \mathcal{A}.$$

(c) Show that

$$V^\star(s) = \max_{a \in \mathcal{A}} Q^\star(s,a), \qquad \forall\, s \in \mathcal{S}.$$

(d) Show that a policy π (not necessarily deterministic) is optimal if and only if

$$\operatorname{supp}\big(\pi(\cdot\,|\,s)\big) \subseteq \operatorname{argmax} Q^\star(s,\cdot), \qquad \forall\, s \in \mathcal{S}.$$

i.e., show that π is optimal if and only if it selects actions that maximize $Q^\star(s,\cdot)$.

(e) Let $\pi^\star$ and $\nu^\star$ be two optimal policies. For any $\theta \in [0,1]$, show that

$$\mu^\star = \theta\pi^\star + (1-\theta)\nu^\star$$

is also an optimal policy. Conclude that the set of optimal policies is a convex set.

Hint. For (d), show that $\mathcal{B}^\pi[V^\star](s) = \mathop{\mathbb{E}}_{a\sim\pi(\cdot\,|\,s)}[Q^\star(s,a)\,|\,s]$.

Problem 5: *Exercise with advantage.* For any policy π, let $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ be the advantage of a at s.

(a) Show that

$$\mathop{\mathbb{E}}_{a\sim\pi(\cdot\,|\,s)}[A^\pi(s,a)\,|\,s] = 0$$

(b) Show that π is optimal if and only if $[A^\pi(s,a) \le 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}]$.

Problem 6: *Removing past rewards from policy gradients* Consider an MDP with no terminal state, i.e. $T = \infty$ with probability 1. Let k be a fixed positive integer. Consider the dynamics where we take actions based on policy π_θ for $t \neq k$, and we take the action based on the policy $\pi_{\theta+\delta}$ at $t = k$.

(a) Show that

$$\nabla_\delta \mathop{\mathbb{E}}_{\substack{s_0\sim p_0\\ a_t\sim\pi_\theta(\cdot\,|\,s_t)\text{ for }t\neq k\\ a_k\sim\pi_{\theta+\delta}(\cdot\,|\,s_k)\\ (r_t,s_{t+1})\sim p(\cdot,\cdot\,|\,s_t,a_t)}} [r_0 + r_1 + r_2 + \cdots + r_{k-1}] = 0.$$

(b) Let $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots)$ be the (continual) trajectory, and let $H(\tau)$ be some function of the trajectory. Show that

$$\nabla_\delta \left(\mathop{\mathbb{E}}_{\substack{s_0\sim p_0\\ a_t\sim\pi_\theta(\cdot\,|\,s_t)\text{ for }t\neq k\\ a_k\sim\pi_{\theta+\delta}(\cdot\,|\,s_k)\\ (r_t,s_{t+1})\sim p(\cdot,\cdot\,|\,s_t,a_t)}} [H(\tau)] \right)\Bigg|_{\delta=0} = \mathop{\mathbb{E}}_{\substack{s_0\sim p_0\\ a_t\sim\pi_\theta(\cdot\,|\,s_t)\text{ for all }t\\ (r_t,s_{t+1})\sim p(\cdot,\cdot\,|\,s_t,a_t)}} [H(\tau)\nabla_\theta \log \pi_\theta(a_k\,|\,s_k)].$$

(c) Show that

$$\mathop{\mathbb{E}}_{\substack{s_0\sim p_0\\ a_t\sim\pi_\theta(\cdot\,|\,s_t)\text{ for all }t\\ (r_t,s_{t+1})\sim p(\cdot,\cdot\,|\,s_t,a_t)}} [(r_0 + r_1 + r_2 + \cdots + r_{k-1})\nabla_\theta \log \pi_\theta(a_k\,|\,s_k)] = 0.$$

Remark. The goal of this problem is to ascribe meaning to the terms in the "enhancement #1" of the policy gradient derivation.

Problem 7: *MMSE estimator.* Let $(X, Y) \sim P$ be a pair of random variables. Assume you have full knowledge of P and you observe Y. However, you did not observe X, and your goal is to estimate the unknown value of X. Your *estimator* is a function of your observed data, and you wish to find the function that minimizes the mean-squared error with respect to X, i.e., we wish to solve

$$\mathop{\text{minimize}}_{f} \quad \mathop{\mathbb{E}}_{(X,Y)\sim P}\big[(X - f(Y))^2\big].$$

A solution $f^\star$ to this optimization problem is the minimum mean square error (MMSE) estimator. Show that

$$f^\star(Y) = \mathop{\mathbb{E}}_{X\sim P_{X\,|\,Y}}[X\,|\,Y].$$

Problem 8: *Pushing up and down probabilities in PG.* Consider an MDP with state space $\mathcal{S} = \{1, \dots, \ell\}$ and action space $\mathcal{A} = \{1, 2, \dots, k\}$. Let $\mu \colon \mathbb{R}^k \to \mathbb{R}^k$ be the softmax function defined as

$$\mu_i(z) = \big(\mu(z)\big)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

for $i = 1, \dots, k$. Let $f_\Theta \colon \mathcal{S} \to \mathbb{R}^k$ be a defined as

$$f_\Theta(s) = \theta_s, \qquad \text{for } s = 1, \dots, \ell,$$

where $\theta_1, \dots, \theta_\ell \in \mathbb{R}^k$ are the trainable parameters. We use the notation

$$\Theta = \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_\ell \end{bmatrix} \in \mathbb{R}^{k \times \ell}.$$

With some abuse of notation, we denote our policy π_Θ as

$$\pi_\Theta(s) = \mu(f_\Theta(s)) = \begin{bmatrix} \mathbb{P}(a = 1 \,|\, s) \\ \mathbb{P}(a = 2 \,|\, s) \\ \vdots \\ \mathbb{P}(a = k \,|\, s) \end{bmatrix}.$$

Let $a_0 \in \mathcal{A}$ and $s_0 \in \mathcal{S}$.

(a) Show that

$$\log \pi_\Theta(a_0 \,|\, s_0) = \Theta_{a_0, s_0} - \log\big(\mathbf{1}^\intercal e^{\theta_{s_0}}\big),$$

where $\mathbf{1} \in \mathbb{R}^k$ is the vector with all entries 1 and $e^{\theta_{s_0}} \in \mathbb{R}^k$ is the element-wise exponentiation of $\theta_{s_0} \in \mathbb{R}^k$.

(b) Show that

$$\nabla_{\theta_{s'}} \log \pi_\Theta(a_0 \,|\, s_0) = 0, \qquad \text{for } s' \neq s_0.$$

(c) Show that

$$\nabla_{\theta_{s_0}} \log \pi_\Theta(a_0 \,|\, s_0) = u_{a_0} - \frac{e^{\theta_{s_0}}}{\mathbf{1}^\intercal e^{\theta_{s_0}}} = u_{a_0} - \pi_\Theta(\cdot \,|\, s_0).$$

where $u_{a_0} \in \mathbb{R}^k$ is the (a_0)-th unit vector with all 0 entries except a 1 in the (a_0)-th coordinate.

(d) Let $g \in \mathbb{R}^k$ such that $g_1 > g_j$ for $j = 2, \dots, k$. Show that

$$\mu_1(z + \alpha g) > \mu_1(z)$$

for sufficiently small $\alpha > 0$.

(e) Let $s_0 \in \mathcal{S}$ and $a_0 \in \mathcal{A}$. Let

$$g = \nabla_\Theta \log \pi_\Theta(a_0 \,|\, s_0).$$

Show that

$$\pi_{\Theta + \alpha g}(a_0 \,|\, s_0) > \pi_\Theta(a_0 \,|\, s_0)$$

for sufficiently small $\alpha > 0$.

(f) Let $s_0 \in \mathcal{S}$. Let

$$g = \mathop{\mathbb{E}}_{a \sim \pi_\Theta(\cdot \,|\, s_0)} \big[C_a \nabla_\Theta \log \pi_\Theta(a \,|\, s_0) \,\big|\, s_0 \big].$$

Assume $C_1 - \mathop{\mathbb{E}}_{a \sim \pi(\cdot \,|\, s_0)} [C_a \,|\, s_0] > 0$ and $C_j - \mathop{\mathbb{E}}_{a \sim \pi(\cdot \,|\, s_0)} [C_a \,|\, s_0] < 0$ for $j = 2, \ldots, k$. Show that

$$\pi_{\Theta + \alpha g}(a = 1 \,|\, s_0) > \pi_\Theta(a = 1 \,|\, s_0)$$

for sufficiently small $\alpha > 0$.

(g) Let $s_0 \in \mathcal{S}$. Let

$$g = \mathop{\mathbb{E}}_{a \sim \pi_\Theta(\cdot \,|\, s_0)} \big[C_a \nabla_\Theta \log \pi_\Theta(a \,|\, s_0) \,\big|\, s_0 \big].$$

Assume $C_1, \ldots, C_k > 0$. Show that it is possible that

$$\pi_{\Theta + \alpha g}(a = 1 \,|\, s_0) < \pi_\Theta(a = 1 \,|\, s_0).$$

for sufficiently small $\alpha > 0$. (Construct a specific example with $k = 2$.)

(h) Let $s_0 \in \mathcal{S}$. Let

$$g = \mathop{\mathbb{E}}_{a \sim \pi_\Theta(\cdot \,|\, s_0)} \big[C_a \nabla_\Theta \log \pi_\Theta(a \,|\, s_0) \,\big|\, s_0 \big].$$

Assume $C_1 > 0$ and $C_2, \ldots, C_k < 0$. Show that it is possible that

$$\pi_{\Theta + \alpha g}(a = 2 \,|\, s_0) > \pi_\Theta(a = 2 \,|\, s_0)$$

for sufficiently small $\alpha > 0$. (Construct a specific example with $k = 3$.)

Problem 9: *Rao–Blackwell theorem with PG.* Consider an MDP with no terminal state, i.e. $T = \infty$ with probability 1. Let π be a policy, not necessarily optimal. Let t be a fixed positive integer. Let

$$(\tau^{(t)}, a_t) = (s_0, a_0, r_0, \cdots, s_{t-1}, a_{t-1}, r_{t-1}, s_t, a_t)$$

be the partial trajectory up to a_t generated by following some policy. Assume the remaining trajectory

$$(r_t, s_{t+1}, a_{t+1}, r_{t+1}, s_{t+2}, a_{t+2}, r_{t+2}, s_{t+3}, \dots)$$

is generated by following by policy π starting from (s_t, a_t). We require $\hat{Q}_t$ to be a random variable such that

$$\mathbb{E}^\pi \left[\hat{Q}_t \,\middle|\, \tau^{(t)}, a_t \right] = Q^\pi(s_t, a_t).$$

(a) Show that

$$\hat{Q}_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots$$

satisfies the requirement.

(b) Show that

$$\hat{Q}_t = r_t + \gamma Q^\pi(s_{t+1}, a_{t+1})$$

satisfies the requirement.

(c) Show that

$$\hat{Q}_t = r_t + \gamma V^\pi(s_{t+1})$$

satisfies the requirement.

(d) Show that

$$\hat{Q}_t = r_t + \gamma r_{t+1} + \gamma^2 V^\pi(s_{t+2})$$

satisfies the requirement.

(e) Show that

$$\mathbb{E}^\pi \Big[\nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \gamma^t \big(\hat{Q}_t - b(s_t) \big) \,\Big|\, \tau^{(t)}, a_t \Big] = \nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \gamma^t \big(Q^\pi(s_t, a_t) - b(s_t) \big)$$

for any $\hat{Q}_t$ satisfying the requirement.

Remark. The Rao–Blackwell theorem stated and proved in class is for scalar random variables, but the Rao–Blackwellized estimator in part (d) is a vector random variable. A vector version of the Rao–Blackwell theorem can be shown with essentially the same steps.

Problem 10: *Rao–Blackwell again.* Consider an MDP with no terminal state, i.e. $T = \infty$ with probability 1. Let π be a policy, not necessarily optimal. Let t be a fixed positive integer. Let the trajectory $(s_0, a_0, r_0, s_1, a_1, r_1, s_2, \dots)$ be generated by policy π.

(a) Let

$$\hat{Q}_t^{\mathrm{TD}(1)} = r_t + \gamma V^\pi(s_{t+1}), \quad \hat{Q}_t^{\mathrm{TD}(2)} = r_t + \gamma r_{t+1} + \gamma^2 V^\pi(s_{t+2}), \quad \hat{Q}_t^{\mathrm{TD}(\infty)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots .$$

Show that

$$\mathbb{E}[\hat{Q}_t^{\mathrm{TD}(1)}] = \mathbb{E}[\hat{Q}_t^{\mathrm{TD}(2)}] = \mathbb{E}[\hat{Q}_t^{\mathrm{TD}(\infty)}]$$

and

$$\mathrm{Var}(\hat{Q}_t^{\mathrm{TD}(1)}) \le \mathrm{Var}(\hat{Q}_t^{\mathrm{TD}(2)}) \le \mathrm{Var}(\hat{Q}_t^{\mathrm{TD}(\infty)}).$$

(b) Let

$$\hat{Q}_t^{\mathrm{TD}(1)} = r_t + \gamma V^\pi(s_{t+1}), \qquad \hat{Q}_t^{\mathrm{TD}(1.5)} = r_t + \gamma Q^\pi(s_{t+1}, a_{t+1})$$

Show that

$$\mathbb{E}[\hat{Q}_t^{\mathrm{TD}(1)}] = \mathbb{E}[\hat{Q}_t^{\mathrm{TD}(1.5)}]$$

and

$$\mathrm{Var}(\hat{Q}_t^{\mathrm{TD}(1)}) \le \mathrm{Var}(\hat{Q}_t^{\mathrm{TD}(1.5)}).$$

Problem 11: *GAE derivations.* Consider an MDP with no terminal state, i.e. $T = \infty$ with probability 1. Let $\gamma \in (0, 1]$. Let π be a policy, not necessarily optimal. Let

$$\delta_t^{V^\pi} = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t), \qquad \text{for } t = 0, 1, \dots .$$

(a) Show that

$$\mathbb{E}[\delta_t^{V^\pi} \mid s_t, a_t] = A^\pi(s_t, a_t), \qquad \text{for } t = 0, 1, \dots .$$

(b) Show that

$$\mathbb{E}[\delta_t^{V^\pi} \mid s_t] = 0, \qquad \text{for } t = 0, 1, \dots .$$

(c) Show that

$$\mathbb{E}[\delta_{t+\ell}^{V^\pi} \mid s_t, a_t] = 0, \qquad \text{for } \ell \ge 1,\ t = 0, 1, \dots .$$

(d) Show that

$$\begin{aligned} \hat{A}_t^{\mathrm{TD}(k)} &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V^\pi(s_{t+k}) - V^\pi(s_t) \\ &= \delta_t^{V^\pi} + \gamma \delta_{t+1}^{V^\pi} + \gamma^2 \delta_{t+2}^{V^\pi} + \cdots + \gamma^{k-1} \delta_{t+k-1}^{V^\pi}, \qquad \text{for } k \ge 1,\ t = 0, 1, \dots . \end{aligned}$$

(e) Let $\lambda \in (0, 1)$. Show that

$$(1 - \lambda)\left(\hat{A}_t^{\mathrm{TD}(1)} + \lambda \hat{A}_t^{\mathrm{TD}(2)} + \lambda^2 \hat{A}_t^{\mathrm{TD}(3)} + \cdots\right) = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V, \qquad \text{for } t = 0, 1, \dots .$$

Problem 12: *Policy evaluation for Q and V.* Consider an MDP with discount factor $\gamma \in (0, 1]$. Let π be a policy, not necessarily optimal. Let $s_0 \sim p_0$, $a_0 \in \pi(\cdot \mid s_0)$, $(r_0, s_1) \sim p(\cdot, \cdot \mid s_0, a_0)$. Assume p_0 assigns positive probability on all states in $\mathcal{S}$. Let

$$\hat{Q} = r_0 + \gamma V^\pi(s_1).$$

Assume V_ϕ is a neural network that can represent arbitrary functions (infinite expressive power).

(a) Show that

$$\mathbb{E}\big[(\hat{Q} - V_\phi(s_0))^2\big]$$

is minimized at $V_\phi = V^\pi$.

(b) Show that

$$\mathbb{E}\big[(\hat{Q} - Q_\phi(s_0, a_0))^2\big]$$

is minimized at $Q_\phi = Q^\pi$.

(c) Show that

$$\mathbb{E}\big[(\hat{Q} - V^\pi(s_0))^2\big] \geq \mathbb{E}\big[(\hat{Q} - Q_\pi(s_0, a_0))^2\big].$$

Remark. Although policy evaluations for the Q- and V-value functions both fit the same quantity $\hat{Q}$, they are different in that the fitting function V_ϕ may only depend on s while Q_ϕ may also depend on a.

Problem 13: *Backpropagating continuous tanh-Gaussian actions.* Let $\mu_\theta(s) \in \mathbb{R}^n$ and $\Sigma_\theta(s) \in \mathbb{R}^{n \times n}$ be neural networks parameterized by $\theta \in \mathbb{R}^P$. Assume $\Sigma_\theta(s)$ is symmetric and strictly positive definite for any $s \in \mathcal{S}$ and $\theta \in \mathbb{R}^P$. Given $s \in \mathcal{S}$, let

$$a = \tanh(z), \qquad z \sim \mathcal{N}(\mu_\theta(s), \Sigma_\theta(s)).$$

Let $\pi_\theta(a \mid s)$ be the implicitly defined probability density function of the random variable $a \in \mathbb{R}^n$. Show that

$$\begin{aligned} z &= \tanh^{-1}(a) \\ \log \pi_\theta(a \mid s) &= -\frac{1}{2} \log \det \Sigma_\theta(s) - \frac{1}{2}(z - \mu_\theta(s))^\intercal \Sigma_\theta^{-1}(s)(z - \mu_\theta(s)) \\ &\quad - \frac{n}{2} \log(2\pi) - \sum_{i=1}^{n} \log(1 - a_i^2). \end{aligned}$$

Problem 14: *PPO clipped surrogate objective.* Let $\ell \geq 0$ and $\varepsilon \in (0, 1)$. Define

$$\mathcal{C}_\varepsilon(\ell, A) = \min\left(\ell A, \operatorname{clip}_{1-\varepsilon}^{1+\varepsilon}(\ell)\, A\right).$$

Show that if $A \geq 0$, then

$$\mathcal{C}_\varepsilon(\ell, A) = \min(\ell, 1 + \varepsilon)\, A$$

and that if $A < 0$, then

$$\mathcal{C}_\varepsilon(\ell, A) = \max(\ell, 1 - \varepsilon)\, A.$$

Problem 15: *Policy iteration.* Implement the policy iteration in the Cliff Walk MDP environment. Perform the policy evaluation step exactly using the linear algebra approach.

Problem 16: *Fitted Monte Carlo policy evaluation for Q.* Implement fitted Monte Carlo policy evaluation in the Cliff Walk MDP environment for the Q-value function. Use the neural network provided in the starter code `CliffWalkQ.py`.

Problem 17: *Fitted k-step TD policy evaluation for Q.* Implement fitted k-step TD policy evaluation in the Cliff Walk MDP environment for the Q-value function. Use the neural network provided in the starter code `CliffWalkQ.py`.

Problem 18: *Implementing policy gradient without k-step TD.* In the undiscounted Cliff Walk MDP, implement the deep policy gradient method without k-step TD. Specifically, implement the following pseudo-code:

while (not converged)
  $g_\theta = 0$, $g_\phi = 0$
  sample trajectory $\tau \sim (p_0, \pi_\theta, p)$
  for $t = 0, 1, \ldots, T-1$
    $\widehat{Q} = r_t + r_{t+1} + r_{t+2} + \cdots + r_{T-1}$
    g_θ `+=` $-\big(\nabla_\theta \log \pi_\theta(a_t \mid s_t)\big)\big(\widehat{Q} - V_\phi(s_t)\big)$
    g_ϕ `+=` $\nabla_\phi \frac{1}{2}\big([\![\widehat{Q}]\!] - V_\phi(s_t)\big)^2$
  end
  update θ and ϕ using g_θ and g_ϕ with an optimizer
end

Problem 19: *GRPO for cliffwalk.* For the Cliff Walk MDP, modify the rewards to keep only the terminal rewards ± 100 and remove the intermediate -1 rewards. Implement GRPO. Do not implement KL penalties.

Problem 20: *Why output projection on MHA?* Consider the standard multi-head self-attention (MHA) layer defined by

$$\underbrace{\text{output}}_{L\times d_{\text{out}}} = \underbrace{\text{concat}(\text{head}_1,\ldots,\text{head}_H)}_{L\times Hd_{\text{head}}} W^O$$
$$\underbrace{\text{head}_h}_{L\times d_{\text{head}}} = \text{Attention}(XW_h^Q, XW_h^K, XW_h^V) \quad \text{for } h=1,\ldots,H,$$

where

$$\text{Attention}(\tilde{Q},\tilde{K},\tilde{V}) = \text{softmax}\big(\frac{\tilde{Q}\tilde{K}^\intercal}{\sqrt{d_{\text{attn}}}}\big)\tilde{V}$$
$$W^O \in \mathbb{R}^{Hd_{\text{head}}\times d_{\text{out}}}, \quad W_h^Q, W_h^K \in \mathbb{R}^{d_{\text{in}}\times d_{\text{attn}}}, \quad W_h^V \in \mathbb{R}^{d_{\text{in}}\times d_{\text{head}}}, \quad X \in \mathbb{R}^{L\times d_{\text{in}}}.$$

Let us call this model MHA1.

Next, consider a variant that we call MHA2.

$$\underbrace{\text{output}}_{L\times d_{\text{out}}} = \text{head}_1 + \cdots + \text{head}_H$$
$$\underbrace{\text{head}_h}_{L\times d_{\text{head}}} = \text{Attention}(XW_h^Q, XW_h^K, XW_h^V) \quad \text{for } h=1,\ldots,H,$$

where

$$\text{Attention}(\tilde{Q},\tilde{K},\tilde{V}) = \text{softmax}\big(\frac{\tilde{Q}\tilde{K}^\intercal}{\sqrt{d_{\text{attn}}}}\big)\tilde{V}$$
$$W_h^Q, W_h^K \in \mathbb{R}^{d_{\text{in}}\times d_{\text{attn}}}, \quad W_h^V \in \mathbb{R}^{d_{\text{in}}\times d_{\text{out}}}, \quad X \in \mathbb{R}^{L\times d_{\text{in}}}.$$

(a) Given an MHA1 model, decompose the rows of W^O as

$$W^O = \begin{bmatrix} W_1^O \\ W_2^O \\ \vdots \\ W_H^O \end{bmatrix} \in \mathbb{R}^{Hd_{\text{head}}\times d_{\text{out}}}$$

such that $W_1^O, W_2^O, \ldots, W_H^O \in \mathbb{R}^{d_{\text{head}}\times d_{\text{out}}}$. Show that if we set the parameters of an MHA2 model as $W_h^V \leftarrow W_h^V W_h^O$ for $h=1,\ldots,H$ and keep all other parameters the same, then the MHA1 and MHA2 models are equivalent, i.e., (MHA1(X) = MHA2(X) for all X).

(b) How many trainable parameters do MHA1 and MHA2 have?

(c) If $d_{\text{in}} = d_{\text{out}} = 512$ and $d_{\text{head}} = 64$, what is the difference in the number of trainable parameters?

Problem 21: *Scaling QK inner products.* Assume that $X \in \mathbb{R}^{L \times d_X}$ is randomly initialized as IID unit Guassians, i.e.,

$$X_{\ell,j} \sim \mathcal{N}(0,1), \qquad \ell \in \{1,\ldots,L\},\, j \in \{1,\ldots,d_X\}$$

independently. Let

$$Q = XW^Q, \qquad K = XW^K,$$

where $W^Q, W^K \in \mathbb{R}^{d_X \times d_K}$.

(a) Assume W^Q and W^K are randomly initialized as IID Gaussians with mean 0 and variance $1/d_X$, i.e., use the LeCun initialization. (So, we are assuming X, W^K, and W^Q are mutually independent.) Show that

$$Q_{\ell,j},\, K_{\ell,j}, \qquad \ell \in \{1,\ldots,L\},\, j \in \{1,\ldots,d_K\},$$

have zero mean, have unit variance, and are uncorrelated.

(b) Let

$$A_{\ell,\ell'} = \frac{q_\ell^\mathsf{T} k_{\ell'}}{\sqrt{d_K}}, \qquad \ell, \ell' \in \{1,\ldots,L\},$$

where

$$Q = \begin{bmatrix} — & q_1^\mathsf{T} & — \\ — & q_2^\mathsf{T} & — \\ & \vdots & \\ — & q_L^\mathsf{T} & — \end{bmatrix} \in \mathbb{R}^{L \times d_K}, \qquad K = \begin{bmatrix} — & k_1^\mathsf{T} & — \\ — & k_2^\mathsf{T} & — \\ & \vdots & \\ — & k_L^\mathsf{T} & — \end{bmatrix} \in \mathbb{R}^{L \times d_K}.$$

Show that

$$\mathbb{E}[A_{\ell,\ell'}] = 0, \qquad \text{for all } \ell,\, \ell'$$
$$\mathbb{E}[(A_{\ell,\ell'})^2] = \begin{cases} 1 & \text{if } \ell \neq \ell' \\ \frac{d_X+2}{d_X} & \text{if } \ell = \ell'. \end{cases}$$

Problem 22: *Bradley–Terry as softmax.* Assume we have data of the form

$$(x, y_A, y_B, z) \in \mathcal{D},$$

where $z = 0$ if $y_A > y_B$ as judged by some reward function and $z = 1$ if $y_A < y_B$. Assume there are no ties between y_A and y_B. Let

$$f_\psi(x, y_A, y_B) = \begin{bmatrix} f_\psi^{(1)}(x, y_A, y_B) \\ f_\psi^{(2)}(x, y_A, y_B) \end{bmatrix} \in \mathbb{R}^2$$

be a neural network parameterized by ψ. Consider fitting f_ψ to solve the 2-class classification task of predicting the value of z given (x, y_A, y_B).

(a) Show that the standard cross-entropy loss is

$$\mathcal{L}(\psi) = \sum_{(x,y_A,y_B,z)\in\mathcal{D}} -(1-z)\log \frac{e^{f_\psi^{(1)}(x,y_A,y_B)}}{e^{f_\psi^{(1)}(x,y_A,y_B)} + e^{f_\psi^{(2)}(x,y_A,y_B)}} - z\log \frac{e^{f_\psi^{(2)}(x,y_A,y_B)}}{e^{f_\psi^{(1)}(x,y_A,y_B)} + e^{f_\psi^{(2)}(x,y_A,y_B)}}$$

(b) Further assume

$$f_\psi(x, y_A, y_B) = \begin{bmatrix} f_\psi^{(1)}(x, y_A, y_B) \\ f_\psi^{(2)}(x, y_A, y_B) \end{bmatrix} = \begin{bmatrix} r_\psi(x, y_A) \\ r_\psi(x, y_B) \end{bmatrix}.$$

Show that $\mathcal{L}(\psi)$ recovers the loss used to train the Bradley–Terry model.

Remark. The conclusion is that Bradley–Terry is the 2-class soft-max regression with a specific parameterization for the neural network.

Problem 23: *Better estimator for KL-divergence.* Let $p(x)$ and $q(x)$ be probability mass functions for $x \in \mathcal{X}$. Then,

$$D_{\mathrm{KL}}(p\|q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \mathop{\mathbb{E}}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right]$$

It is well known that $D_{\mathrm{KL}}(p\|q) \geq 0$, and the proof follows from an application of Jensen's inequality. Assume we have data $X_1, \ldots, X_N \sim p$.

(a) Show that

$$\hat{D}^{(1)} = \frac{1}{N}\sum_{i=1}^{N} \log \frac{p(X_i)}{q(X_i)}$$

is an unbiased estimator of $D_{\mathrm{KL}}(p\|q)$.

(b) Show that $\hat{D}^{(1)} < 0$ is possible.

(c) Show that

$$\hat{D}^{(2)} = \frac{1}{N}\sum_{i=1}^{N} \left(\frac{q(X_i)}{p(X_i)} - \log \frac{q(X_i)}{p(X_i)} - 1\right)$$

is an also unbiased estimator of $D_{\mathrm{KL}}(p\|q)$.

(d) Show that $\hat{D}^{(2)} \geq 0$ always holds.

Remark. The original InstructGPT uses $\hat{D}^{(1)}$ to estimate the KL-penalty, but many subsequent works, such as the GRPO paper, use $\hat{D}^{(2)}$.

Problem 24: *Encoder-only transformers without positional embeddings are permutation equivariant.* Let σ be a permutation of length L, i.e., $\sigma(1), \sigma(2), \ldots, \sigma(L)$ take values $1, \ldots, L$ exactly once. If

$$x_1, x_2, \ldots, x_L$$

is a sequence of tokens,

$$x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(L)}$$

is the permuted (shuffled) sequence of tokens. Let f be an encoder-only transformer. Specifically, f is a composition of the token embedding layer with multiple Pre-LN transformer layers without the causal mask. For simplicity, let us ignore the tokenizer and view f as a function of the one-hot tokens $u_1, \ldots, u_L \in \mathbb{R}^N$ and

$$f(u_1, \ldots, u_L) = (y_1, \ldots, y_L),$$

where $y_1, \ldots, y_L \in \mathbb{R}^d$. For simplicity, do not consider an output embedding layer or a classification head. Crucially, assume positional embeddings are not used. Show that

$$f(u_{\sigma(1)}, \ldots, u_{\sigma(L)}) = (y_{\sigma(1)}, \ldots, y_{\sigma(L)}),$$

i.e., if the input is shuffled, the output is exactly the same but shuffled in the same way.

Remark. This property is referred to as permutation equivariance.

Problem 25: *Inferring absolute position with NoPE.* Consider a token embedding layer and a masked single-head self-attention layer mapping $\{u_\ell\}_{\ell=1}^L \mapsto \{x_\ell\}_{\ell=1}^L \mapsto \{y_\ell\}_{\ell=1}^L$ as

$$
\begin{aligned}
x_\ell &= M u_\ell \qquad \text{for } \ell = 1, \dots, L \\
q_\ell = (W^Q)^\intercal x_\ell, \qquad k_\ell &= (W^K)^\intercal x_\ell, \qquad v_\ell = (W^V)^\intercal x_\ell \qquad \text{for } \ell = 1, \dots, L \\
\tilde{A}_{ij} &= \begin{cases} q_i^\intercal k_j / \sqrt{d_K} & \text{if } i \geq j \\ -\infty & \text{if } i < j \end{cases} \qquad \text{for } i, j \in \{1, \dots, L\} \\
A_{ij} &= \frac{e^{\tilde{A}_{ij}}}{\sum_{j'=1}^L e^{\tilde{A}_{ij'}}}, \qquad \text{for } i, j \in \{1, \dots, L\} \\
y_\ell &= \sum_{r=1}^{\ell} A_{\ell r} v_r, \qquad \text{for } \ell = 1, \dots, L,
\end{aligned}
$$

where $u_1, \dots, u_L \in \mathbb{R}^N$ are the tokenized one-hot vectors, $M \in \mathbb{R}^{d \times N}$ represents the token embedding layer, $\{u_\ell\}_{\ell=1}^L \subset \mathbb{R}^N$, $\{x_\ell\}_{\ell=1}^L \subset \mathbb{R}^d$, $\{y_\ell\}_{\ell=1}^L \subset \mathbb{R}^d$, $A \in \mathbb{R}^{L \times L}$ contain the attention weights, $(W^Q)^\intercal, (W^K)^\intercal \in \mathbb{R}^{d_K \times d}$, and $(W^V)^\intercal \in \mathbb{R}^{d \times d}$. In particular, no explicit positional embeddings are used. Assume the message starts with the special token

$$u_1 = \texttt{<|im\_start|>},$$

and, without loss of generality, assume `<|im_start|>` is first token, i.e., $u_1 = e_1$, where $e_1 \in \mathbb{R}^N$ is the unit vector with a 1 in the first coordinate and 0's everywhere else. Let

$$
M = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \cdots & * \end{bmatrix}, \qquad (W^K)^\intercal = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}, \qquad (W^V)^\intercal = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \cdots & * \end{bmatrix},
$$

where $*$ denotes an arbitrary value. Let $(W^Q)^\intercal \in \mathbb{R}^{d_K \times d}$ be arbitrary. Show that

$$
y_\ell = \begin{bmatrix} 1/\ell \\ * \\ \vdots \\ * \end{bmatrix} \qquad \text{for } \ell = 1, \dots, L.
$$

Remark. This problem shows that there is a configuration of the transformer such that the inverse of the absolute position is revealed in the first coordinates of $y_1, \dots, y_L$, even though no explicit positional embedding mechanism was used. This result also shows that a masked single-head self-attention layer (and therefore a decoder-only transformer) is not permutation equivariant.

Problem 26: *Softmax bottleneck.* Consider a decoder-only transformer with an output projection layer that maps

$$v_1, \ldots, v_L \in \mathbb{R}^h$$

to

$$w_1, \ldots, w_L \in \mathbb{R}^N$$

with

$$w_\ell = Bv_\ell \qquad \text{for } \ell = 1, \ldots, L,$$

where h is the hidden dimension and N is the number of tokens. After this, the output probabilities will be computed via $\mu(w_\ell)$, where μ is the softmax function defined by

$$\big(\mu(z)\big)_i = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \qquad \text{for } i = 1, \ldots, N.$$

Throughout this problem, use the notation

$$B = \begin{bmatrix} — b_1^\intercal — \\ \vdots \\ — b_N^\intercal — \end{bmatrix} \in \mathbb{R}^{N \times h},$$

so $b_1, \ldots, b_N \in \mathbb{R}^h$. Assume $h < N - 1$, as is the case in practice.

This standard setup is potentially problematic because $\mu(w_\ell) = \mu(Bv_\ell)$ cannot possibly represent an arbitrary probability distribution on N tokens (which has $N - 1$ degrees of freedom), because $v_\ell \in \mathbb{R}^h$ only has h degrees of freedom. This limitation is called the *softmax bottleneck.*

(a) Assume that the rows of B are normalized and distinct, i.e.,

$$\|b_k\| = 1 \qquad \text{for } k = 1, \ldots, N$$

and

$$b_k \neq b_{k'} \qquad \text{for } k \neq k'.$$

Show that for any unit vector $e_k \in \mathbb{R}^N$ (which is the one-hot vector with non-zero component at position k), there is a v_ℓ such that

$$\mu(Bv_\ell) \approx e_k,$$

where $\approx$ can be made as accurate as we want it to be.

(b) As a toy example, let $h = 2$, $N = 8$, and

$$b_k = \begin{bmatrix} \cos(\pi k/4) \\ \sin(\pi k/4) \end{bmatrix} \qquad \text{for } k = 1, \ldots, 8.$$

Show that there is a v_ℓ such that

$$\mu(Bv_\ell) \approx (1/2, 1/2, 0, 0, 0, 0, 0, 0).$$

(c) Consider the setup of part (b). Show that

$$\mu(Bv_\ell) \approx (1/2, 0, 1/2, 0, 0, 0, 0, 0)$$

is not possible.

Remark. The takeaway is that despite the softmax bottleneck, one-hot vectors can be represented as the output distribution. However, some distributions where multiple tokens share the probabilities may not be representable.

Problem 27: *Parameter and FLOP count of transformers.* Consider a multi-head self-attention (MHA) layer without the causal mask, followed by a positionwise FFN with expansion factor 4. Specifically, the operation maps $\{x_\ell\}_{\ell=1}^L \mapsto \{w_\ell\}_{\ell=1}^L$ as

$$\begin{aligned}
&x_1, \dots, x_L \in \mathbb{R}^d, \qquad \{x_\ell\}_{\ell=1}^L = X \in \mathbb{R}^{L\times d} \\
&\text{for } h = 1, \dots, H \\
&\qquad Y_h = \text{Attention}(XW_h^Q, XW_h^K, XW_h^V) \in \mathbb{R}^{L\times d_K} \\
&Z = \text{MHA}(X) = \text{concat}(Y_1, \dots, Y_H)W^O \\
&z_1, \dots, z_L \in \mathbb{R}^d, \qquad \{z_\ell\}_{\ell=1}^L = Z \in \mathbb{R}^{L\times d} \\
&w_\ell = W_2\sigma(W_1 z_\ell), \qquad \text{for } \ell = 1, \dots, L
\end{aligned}$$

where $W_1 \in \mathbb{R}^{4d\times d}$ and $W_2 \in \mathbb{R}^{d\times 4d}$, σ is some activation function and the single-head attention layer is defined as

$$\begin{gathered}
Q = XW^Q \in \mathbb{R}^{L\times d_K}, \qquad K = XW^K \in \mathbb{R}^{L\times d_K}, \qquad V = XW^V \in \mathbb{R}^{L\times d_K} \\
Y = \text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\intercal}{\sqrt{d_K}}\Big)V \in \mathbb{R}^{L\times d_K} \\
A_{ij} = \frac{e^{q_i^\intercal k_j/\sqrt{d_K}}}{\sum_{j'=1}^L e^{q_i^\intercal k_{j'}/\sqrt{d_K}}}, \qquad \text{for } i, j \in \{1, \dots, L\} \\
y_\ell = \sum_{r=1}^L A_{\ell r} v_r, \qquad \text{for } \ell = 1, \dots, L \\
y_1, \dots, y_L \in \mathbb{R}^{d_K}, \qquad \{y_\ell\}_{\ell=1}^L = Y \in \mathbb{R}^{L\times d_K}.
\end{gathered}$$

Note that q-, k-, and v-vectors share the same dimension d_K. Finally, set $d_K = d/H$, as is commonly done in modern transformers.

(a) Show that the trainable parameters in the MHA and the position-wise FFN layers are roughly comparable.

(b) Show that the number of necessary arithmetic operations required to do a forward pass (computing $\{x_\ell\}_{\ell=1}^L \mapsto \{w_\ell\}_{\ell=1}^L$) is on the order of

$$\Theta\big(L^2 d + Ld^2\big)$$

(c) In the FLOP estimate of (b), at what value of L does the first term (dependent on L^2) become more dominant?

Remark. For references, the Llama 3 405B model has dimensions $d = 16384$ and $H = 128$.

Remark. The key takeaway is that the inference cost of LLMs does not scale quadratically with the sequence length L for moderate values of L, despite some incorrect claims to the contrary in the literature. Moreover, as L increases, compute efficiency improves, making the L^2 term less visible in practical timing measurements.

Problem 28: *Unbiasedness of Dr. GRPO.* In this problem, we derive the unbiasedness property of Dr. GRPO. To clarify, the statement [the "advantage" estimate of Dr. GRPO is an unbiased estimate of the advantage function] is not true. Rather, the claim of correctness and "unbiasedness" of the Dr. GRPO is made through the following analysis.

(a) Let

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, \dots)$$

be a trajectory of an MDP. Let Y be a random variable independent of τ. Show that

$$\mathop{\mathbb{E}}_{\substack{\tau \sim (p_0, \pi_\theta, p) \\ Y}} [\nabla_\theta \log \pi_\theta(a_t \mid s_t) b(Y)] = 0.$$

(b) Consider an undiscounted MDP ($\gamma = 1$) with terminal rewards as in the GRPO setup. Let $\tau^{(1)}, \dots, \tau^{(N)}$ be IID trajectories with terminal times $T^{(1)}, \dots, T^{(N)}$ and terminal rewards $r^{(1)}, \dots, r^{(N)}$. To clarify, for $i = 1, \dots, N$,

$$\tau^{(i)} = \big(s_0^{(i)}, a_0^{(i)}, s_1^{(i)}, a_1^{(i)}, \dots, s_{T^{(i)}-1}^{(i)}, a_{T^{(i)}-1}^{(i)}, r^{(i)}, s_{T^{(i)}}^{(i)}\big),$$

where $s_{T^{(i)}}^{(i)} =$ `<term>` and the rewards at all times except the terminal one are all 0. Let

$$\mathrm{mean}(\mathbf{r}) = \frac{1}{N} \sum_{i=1}^{N} r^{(i)}.$$

Show that

$$\mathop{\mathbb{E}}_{\tau^{(1)}, \dots, \tau^{(N)}} \left[\frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T^{(i)}-1} \nabla_\theta \log \pi_\theta(a_t^{(i)} \mid s_t^{(i)}) \big(r^{(i)} - \mathrm{mean}(\mathbf{r})\big) \right] = \frac{N-1}{N} \nabla \mathcal{J}(\theta).$$

To clarify, in this undiscounted terminal reward MDP setup,

$$\mathcal{J}(\theta) = \mathop{\mathbb{E}}_{\tau \sim (p_0, \pi_\theta, p)} \big[r\big],$$

where r is the terminal reward of the trajectory $\tau \sim (p_0, \pi_\theta, p)$.

Remark. Replacing the advantage estimate in policy-gradient-type methods with $r^{(i)} - \mathrm{mean}(\mathbf{r})$ was previously explored under the name *REINFORCE Leave-One-Out (RLOO)*, although it did not receive much mainstream attention before GRPO and Dr. GRPO. Also, the now-common practice of referring to $r^{(i)} - \mathrm{mean}(\mathbf{r})$ as an "advantage estimate" is somewhat misleading, as it is not an unbiased estimate of the true advantage function.