Cornell University
CS 5788: Introduction to Generative Models
Spring 2026 Instructor: Andrew Owens

Problem Set 4: Language Models

Posted: Tuesday, April 7, 2026	**Due:** Tuesday, May 5, 2026

Please submit your written solution to Gradescope as a `.pdf` file. Please convert your Colab notebooks to PDF. For your convenience, we have included the PDF conversion script at the end of the notebook.

We recommend editing and running your code in Google Colab, although you are welcome to use your local machine instead. Please note that problems marked optional will not be graded.

Problem 4.1 *Decoder-only Transformer*

We will implement the decoder-only transformer and use it to generate text. The architecture closely follows that of the original paper (Attention Is All You Need). The notebook `decoder-only-transformer.ipynb` will walk you through the implementation.

Problem 4.2 *Supervised Finetuning and Direct Preference Optimization*

We will implement the (DPO) algorithm. The notebook `sft-dpo.ipynb` will walk you through the implementation.

Problem 4.3 *Written problems*

Please turn in your answers (either written or typeset) as a separate PDF.

(a) In this problem, we will justify the teacher-forcing objective used to train autoregressive language models.

Let x be an input prompt and let $y = (y_1, \ldots, y_T)$ be a target output sequence drawn from the dataset $\mathcal{D}$. An autoregressive language model can be defined as

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}), \tag{1}$$

where $y_{<t} = (y_1, \ldots, y_{t-1})$.

The standard training objective is the teacher-forcing objective

$$\mathcal{L}_{\text{TF}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t}) \right]. \tag{2}$$

(i) Show that minimizing $\mathcal{L}_{\mathrm{TF}}(\theta)$ in Equation 2 is equivalent to maximum likelihood estimation of the full conditional sequence model $\pi_\theta(y \mid x)$ in Equation 1.

(ii) Now consider a hypothetical alternative training procedure where, at each step, the model conditions on its *own previously generated tokens*

$$\hat{y}_{<t} \sim \pi_\theta(\cdot \mid x),$$

and is trained with the objective

$$\mathcal{L}_{\mathrm{self}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, \hat{y}_{<t})\right]. \tag{3}$$

Briefly explain why this objective is *not* a good maximum likelihood estimation for the data distribution. Based on your reasoning, briefly justify why teacher forcing remains the better objective for MLE, even though at test time the model will generate tokens autoregressively using its own previous outputs.

(b) In this problem, we will derive the policy-gradient objective used in reinforcement learning for language models. Consider a language model policy $\pi_\theta(y \mid x)$, where x is a prompt and y is a generated completion. We view $\pi_\theta(\cdot \mid x)$ as a distribution over completions conditioned on the prompt. Let $r(x, y)$ be a scalar reward assigned to the completion.

We define the reward-maximization objective as

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[r(x, y)]. \tag{4}$$

In practice, this objective appears in reinforcement learning fine-tuning of language models, where the reward may come from task success, human preference, a reward model, or a verifier.

(i) Use the log-derivative trick

$$\nabla_\theta \pi_\theta(y \mid x) = \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x) \tag{5}$$

to show that

$$\nabla_\theta J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[r(x, y) \nabla_\theta \log \pi_\theta(y \mid x)\right]. \tag{6}$$

This is the REINFORCE gradient estimator.

(ii) A common variance-reduction technique is to subtract a baseline $b(x)$ that depends only on the prompt. Consider the estimator

$$\nabla_\theta J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[(r(x, y) - b(x)) \nabla_\theta \log \pi_\theta(y \mid x)\right]. \tag{7}$$

Show that this estimator is still unbiased, i.e., it has the same expectation as REINFORCE in Equation 6.

(c) In this problem, we study preference-based post-training of language models. We will first derive the maximum-likelihood objective for reward-model training under the Bradley–Terry model, and then derive the Direct Preference Optimization (DPO) objective from a KL-regularized reinforcement-learning formulation.

Suppose we are given a dataset of preference comparisons of the form (x, y^+, y^-), where for prompt x, the response y^+ is preferred over y^-. Let $r_\phi(x, y)$ denote a learned scalar reward model.

(i) **Bradley–Terry preference model.**

In the Bradley–Terry model, the probability that y^+ is preferred to y^- is parameterized as

$$P_\phi(y^+ \succ y^- \mid x) = \frac{\exp(r_\phi(x, y^+))}{\exp(r_\phi(x, y^+)) + \exp(r_\phi(x, y^-))}. \tag{8}$$

(a) Show that this can be rewritten as

$$P_\phi(y^+ \succ y^- \mid x) = \sigma\big(r_\phi(x, y^+) - r_\phi(x, y^-)\big)\,, \tag{9}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid.

(b) Given a dataset

$$\mathcal{D}_{\text{pref}} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N, \tag{10}$$

write down the likelihood of the dataset under the model, assuming all comparisons are independent.

(c) Derive the following negative log-likelihood objective that one would minimize to train the reward model:

$$\mathcal{L}_{\text{RM}}(\phi) = -\sum_{i=1}^{N} \log \sigma\big(r_\phi(x_i, y_i^+) - r_\phi(x_i, y_i^-)\big)\,. \tag{11}$$

(d) Show that if we replace the reward model $r_\phi(x, y)$ by

$$\tilde{r}_\phi(x, y) = r_\phi(x, y) + c(x), \tag{12}$$

where $c(x)$ is any function that depends only on the prompt x and not on the completion y, then the Bradley–Terry preference probability remains unchanged. In other words, for each prompt x, adding the same constant to all candidate completions does not change any pairwise preference probabilities, so the data can only identify rewards up to an additive offset.

(ii) **Deriving DPO from KL-regularized RL.**

We now derive the DPO loss. Let $\pi_{\text{ref}}(y \mid x)$ be a fixed reference policy, and let $\pi(y \mid x)$ be the policy we want to optimize. Consider the KL-regularized RL objective

$$\max_{\pi}\ \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi(\cdot|x)}[r(x, y)] - \beta\, D_{\text{KL}}(\pi(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x))\right], \tag{13}$$

where $\beta > 0$ is a regularization parameter.

(a) Fix a prompt x. Expand the KL divergence in Equation 13 and show that the inner optimization problem can be written as

$$\max_{\pi(\cdot|x)} \sum_y \pi(y \mid x) \left[r(x, y) - \beta \log \frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)}\right]. \tag{14}$$

(b) **(Optional)** Derive the optimal policy $\pi^*(y \mid x)$ of the objective in Equation 13 and show that

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right), \tag{15}$$

where $Z(x)$ is a partition function

$$Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right). \tag{16}$$

(c) Based on Equation 15, show that the optimal policy implies the reward can be written as

$$r(x, y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \tag{17}$$

(d) Now suppose that preferences are generated according to the Bradley–Terry model applied to the *optimal* reward:

$$P(y^+ \succ y^- \mid x) = \sigma\big(r(x, y^+) - r(x, y^-)\big)\,. \tag{18}$$

Substitute the expression from part (c) into this probability and show that the additive term $\beta \log Z(x)$ cancels.

(e) Replacing π^* by a parameterized policy π_θ, derive the DPO objective for a preference dataset

$$\mathcal{D}_{\text{pref}} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N. \tag{19}$$

by showing that maximizing the likelihood of the preferences is equivalent to minimizing

$$\mathcal{L}_{\text{DPO}}(\theta) = -\sum_{i=1}^{N} \log \sigma\left(\beta \left[\log \frac{\pi_\theta(y_i^+ \mid x_i)}{\pi_{\text{ref}}(y_i^+ \mid x_i)} - \log \frac{\pi_\theta(y_i^- \mid x_i)}{\pi_{\text{ref}}(y_i^- \mid x_i)}\right]\right). \tag{20}$$

Credits. The problem set was written by Zhaolin Gao and Andrew Owens.