Quiz 6

601.471/671 NLP: Self-Supervised Models

Johns Hopkins University Fall 2026

Name: _____ **Instructor version**

Time: 20 minutes. Total: 100 points.

Give short answers and show calculations. No calculator is needed; log is the natural logarithm.

Q1. FFNs and Block Layouts *30 points*

(a) **15 points. FFN budgets.** Compare bias-free FFNs on a row vector $x \in \mathbb{R}^D$:

$$\text{ordinary: } f(xW_1)W_2, \qquad \text{gated: } \big(f(xU) \odot (xV)\big)W_o.$$

Here $W_1 : [D, F]$, $W_2 : [F, D]$, $U, V : [D, G]$, and $W_o : [G, D]$. All matrices are separately trainable; f is parameter-free; $\odot$ multiplies elementwise.

Review: Parameter counting from Quiz 1, Q3(b). Gated FFNs and matching their budgets are new applications.

(i) **3 points.** Give the ordinary FFN's matrix-parameter count in terms of D, F.

Solution: $DF + FD = \boxed{2DF}$ matrix parameters.

(ii) **3 points.** Give the gated FFN's matrix-parameter count in terms of D, G.

Solution: $DG + DG + GD = \boxed{3DG}$ matrix parameters.

(iii) **6 points.** With $D = 6, F = 24$, find the gated width G that matches the ordinary FFN's parameter count.

Solution: Equate $2DF = 3DG$, giving $G = 2F/3 = \boxed{16}$. Award 3 points for the matching relation and 3 for the correct width.

(iv) **3 points.** For those widths, find the common matrix-parameter count.

Solution: $2(6)(24) = 3(6)(16) = \boxed{288}$. Accept consistent arithmetic follow-through. Matched parameter counts do not guarantee matched runtime or quality.

(b) **15 points. Serial or parallel.** A is attention, M an FFN, and N, N_1, N_2 normalization:

$$\text{serial: } h = x + A(N_1(x)), \quad y = h + M(N_2(h)); \qquad \text{parallel: } y = x + A(N(x)) + M(N(x)).$$

(i) **3 points.** Which layout's FFN receives a state containing this block's attention update?

Solution: The **serial** layout: $h = x + A(N_1(x))$ contains the attention update.

(ii) **3 points.** Write the input to the serial FFN.

Solution: $\boxed{N_2(h)}$.

(iii) **3 points.** Write the input to the parallel FFN.

Solution: $\boxed{N(x)}$.

(iv) **6 points.** Name one computation that the parallel branches can share or combine.

Solution: Reuse one computation of $N(x)$ for both parallel branches. Alternatively, combine their input projections in one matrix multiplication and split the features afterward. One valid example suffices; a speedup depends on implementation and hardware.

Q2. Pre-training Data and Optimization *30 points*

(a) **10 points. Held-out data.** A pipeline randomly splits pages into training and validation, then removes exact duplicate documents *within each split* separately.

(i) **4 points.** Why can identical document text still appear in both splits?

Solution: One copy can land in each split and survive because within-split checks never compare the two copies.

(ii) **6 points.** Give one pipeline change that prevents this exact overlap while keeping held-out validation.

Solution: Group pages by exact text before splitting and assign each entire group to one split. Alternatively, freeze validation and remove training documents that match it. A valid fix must prevent the exact overlap and preserve held-out validation; near-duplicates and copied fragments need additional checks.

(b) **10 points. Fixed compute.** These synthetic runs have comparable architecture family, tokenizer, training mixture, validation set, and suitable optimizer settings at each size. Measured training compute is equal. Lower validation loss is better.

Run	Parameters	Processed tokens	Validation loss
A	100M	8B	3.4
B	200M	4B	3.1
C	400M	2B	3.3

(i) **4 points.** Which measured run is best?

Solution: B: its validation loss, 3.1, is lowest.

(ii) **6 points.** Why would extending only C to 4B tokens, at additional compute cost, change the fixed-compute comparison?

Solution: Extending C spends more compute, so C no longer shares A/B's displayed budget. Equal token counts and equal compute are different comparisons. These runs do not establish a precise or universal optimum.

(c) **5 points. Optimizer memory.** AdamW stores two moment tensors, each with one entry per trainable scalar parameter. A model has 100 million parameters and each moment entry occupies 4 bytes. Count only these tensors; $1\,\text{GB} = 10^9$ bytes.

(i) **5 points.** Find the combined storage of both moment tensors in decimal GB.

Review: Tensor-memory counting from Quiz 5, Q2(a)(iv)–(v), applied here to optimizer state.

Solution: $2 \cdot 100 \cdot 10^6 \cdot 4 = 800 \cdot 10^6$ bytes $= \boxed{0.8\,\text{GB}}$. Award 3 points for $2 \cdot 100 \cdot 10^6 \cdot 4$ bytes and 2 for the conversion to 0.8 GB. Exclude weights, gradients, activations, and other training state.

(d) **5 points. Learning-rate schedules.** A warmup–stable–decay schedule first increases the learning rate, then holds it steady, then reduces it.

(i) **2 points.** Give one reason to use a smaller learning rate at the very start.

Solution: It limits early update magnitudes and can help avoid instability while representations and optimizer state are settling. Warmup does not make every peak rate safe.

(ii) **3 points.** Why might an extremely small learning rate for the *entire* fixed-budget run be a poor choice?

Solution: Learning can be too slow, wasting the finite compute budget. Credit an explanation connecting slow progress to underuse of the fixed training budget.

Q3. Instruction Tuning *20 points*

(a) **20 points. Response-only loss.** An instruction/context consists of tokens P_1, P_2; its demonstrated response is R_1, R_2, EOS. EOS marks the end of the response and is included in its loss. The table shows already aligned next-token predictions; each probability is for the listed target given the *gold* prefix.

Prediction after	P_1	P_2	R_1	R_2
Target token	P_2	R_1	R_2	EOS
Probability of target	$1/4$	$1/2$	$1/4$	$1/2$

(i) **8 points.** Which target tokens in the table should count toward the instruction-tuning loss? List them.

Solution: $\boxed{R_1,\ R_2,\ \text{EOS}}$. Only response targets count toward this loss; the prompt target P_2 is excluded. Award 2 points for each response target and 2 for excluding P_2. Prompt inputs still condition the response.

(ii) **12 points.** Using the target tokens you selected in (i), compute the mean negative log-likelihood in terms of $\log 2$.

Review: Mean negative log-likelihood from Quiz 2, Q2(a), and averaging over selected targets from Quiz 5, Q3(ii). The response-only instruction-tuning setup is new.

Solution: $\ell = -\dfrac{\log(1/2) + \log(1/4) + \log(1/2)}{3} = \boxed{\dfrac{4\log 2}{3}}$. Award 4 points for the response-only numerator, 4 for denominator 3, and 4 for the value. Accept consistent follow-through from a different target selection.

Q4. Learning a Reward Model *20 points*

(a) **5 points. Tokens as actions.** Consider the token-level RL description of response generation in the lecture.

(i) **2 points.** What is the state just before choosing the next token?

Solution: The prompt plus the response prefix already generated.

(ii) **2 points.** What is the action?

Solution: The next token.

(iii) **1 point.** When does the response-level reward arrive in this setup?

Solution: After the response is complete in the lecture's setup. No policy-update derivation is required.

(b) **15 points. Preference probability and loss.** For a prompt p, humans prefer response w over l. The reward model uses

$$P(w \succ l \mid p) = \sigma(\Delta), \qquad \Delta = R(w;p) - R(l;p), \qquad \sigma(z) = \frac{1}{1+e^{-z}}.$$

The example's loss is $-\log \sigma(\Delta)$. Let $R(w;p) = \log 3$ and $R(l;p) = 0$. Use exact expressions.

(i) **6 points.** Find the preference probability.

Solution: $\Delta = \log 3$, so $\sigma(\log 3) = 1/(1+1/3) = \boxed{3/4}$. Award 2 points for the score difference, 2 for substituting into the sigmoid, and 2 for the probability.

(ii) **6 points.** Find this example's loss.

Review: Negative-log-probability calculation from Quiz 2, Q2(a), applied here to a preference probability.

Solution: $-\log(3/4) = \boxed{\log(4/3)}$. Award 3 points for applying negative log to the preference probability and 3 for the correct exact expression. Accept consistent follow-through from (i).

(iii) **3 points.** If the two scores are swapped, does the loss increase or decrease?

Solution: The loss **increases**. Now $\Delta = -\log 3$, the observed preference has probability $1/4$, and its loss is $\log 4$. No decimal approximation is needed.