MCB128

Quiz b1

February 13, 2026

Name: ______________________________ **Score:** ____________

Instructions:

- Show all work for full credit.
- Can check notes; no prompts to chatbot allowed

---

1. (**2/10 pts**) In the first layer of a neural network that uses the ReLU activation function

   $ReLU(a) = \begin{cases} a & if\, a \geq 0 \\ 0 & if\, a < 0 \end{cases}$

   what would happen if we initialize all the weights and biases to zero?

   Any input values $(x_1 \dots x_I)$ will produce a zero output.

   $y_j = ReLU(\sum_i x_i \times W_{ij} + b_j) = ReLU(\sum_i x_i \times 0 + 0) = ReLU(0) = 0,$

2. (**2/10 pts**) (a) Why do we convert the outputs of a neural network (the logits) to a softmax? (b) Why do logits get exponentiated in a softmax?

   (a) So that the output becomes a probability distribution $(p_1, \dots, p_K)$ for a categorical variable of dimension K.

   $$p_k = softmax(y)_k = \frac{e^{y_k}}{\sum_{k'=1}^{K} e^{y_{k'}}}, \quad \sum_k p_k = 1.$$

   Thus, p_k can be interpreted as the probability that category "k" is the correct classification.

   (b) So that the probabilities do not become negatives.

   One could have consider to normalize the logits directly as

   $$q_k = \frac{y_k}{\sum_{k'=1}^{K} y_{k'}}$$

   The normalization condition is guaranteed here $\sum_k q_k = 1$.

   But the other condition of a probability distribution is that all values have to be in the range [0,1].

   The logits y_k are unconstrained values that could be positive or negative, so the above expression could produce negative probabilities. The exponentiation converts all logits into positive numbers, which can them be safely normalized, regardless of the sign of the individual logits.

3. (**2/10 pts**) A neural network uses a training set of 100 examples, and a bachsize of 20. If the total number of stochastic gradient descent iterations is 1,000, for how many epochs was the network trained?

The number of batches per epoch is $100/20 = 5\,\text{batches/epoch}$.

Since each iteration involves one batch, then

$$1000\,\text{its} = 1000\,\text{batches} = 5\,\text{batches/epoch} \times n\,\text{epochs}$$

thus

$$n\,\text{epochs} = \frac{1000}{5} = 200\,\text{epochs}$$

4. (**4/10 pts**) a) Describe in detail all the weights and their dimensions for a complete fully connected network with 2 hidden layers (with H1 neurons in layer 1 and H2 neurons in layer 2), one input layer (with dimension I), and one output layer (with dimension O). (b) If you use one-hot encoding for your inputs (DNA or RNA) and each input has length L and C categories, how can you flatten your input for the input layer of your network? (c) Choose and describe the remaining parameters of the model that you need for training.

(a) First layer

$$W_1[I, H1], \quad b_1[H1]$$

Second layer

$$W_2[H1, H2], \quad b_2[H2]$$

Third layer

$$W_3[H2, O], \quad b_3[O]$$

(b) $I = L \times C$

(c) i. activation functions for the three layers

ii. labels

iii. loss function

iv. optimizer

v. learning rate