Quiz 4
CMSC 421: Introduction to Artificial Intelligence

Instructions:
You may use blank space anywhere (front or back) for scratch work if necessary, but your final answer must be clearly circled under the corresponding problem.

Time Limit: 15 minutes

DID (e.g., jdoe123): ____________________________

Date: ____________________________________________

Honor Pledge:

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination.

Signature: ____________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

1. (1 point) In the context of SGD training of feed-forward neural networks—where each processing pass finds Cost $C$ and then its weight-gradient for a mini-batch (e.g., by backprop)—what is an epoch?
   A. One processing pass through all the examples in a mini-batch
   B. One processing pass through all the examples in the training set
   C. The reduction of Cost by $\frac{1}{2}$ of the initial Cost, or by $\frac{1}{2}$ of the Cost at the end of the previous epoch
   D. None of the above

2. (1 point) The gradient descent algorithm will always narrow in on a local minimum of the Cost function, for all values of alpha (or learning rate, step size).
   A. True  B. False

3. (1 point) The Hadamard product of vectors $(a, b, c)$ and $(d, e, f)$ is
   A. equal to the dot product
   B. $(abc, def)$
   C. $(ad, be, cf)$
   D. $(2ad, 2be, 2cf)$
   E. None of the above

4. (1 point) Given a dataset of size $N$, what is the minimum size of a mini-batch?
   A. $\sqrt{N}$  B. $\lfloor N/2 \rfloor$  C. 100  D. 1

5. (6 points) Below is a feedforward neural network. There are NO BIAS UNITS. The $i$-th unit in layer $l$ is labeled $U^l_i$. As you can see below, the NN receives inputs $x_1, x_2,$ and $x_3,$ and computes two outputs, $a_1$ and $a_2$, which are the activations of $U^1_1$ and $U^1_2$, respectively. Activations of $U^0_i$ are $x_i$, and the activation function for $U^1_i$ is ReLU. The weights are contained in a $3 \times 2$ matrix $W$, where $W_{ij}$ is the weight of the connection from $U^0_i$ to $U^1_j$.

Explain why this neural network cannot learn to output $(y_1, y_2)$ for input $(0, 0, 0)$ if $(y_1, y_2) \neq (0, 0)$.