CS 4700, Foundations of Artificial Intelligence
Spring 2020
Solutions to Quiz 6

1. This had multiple questions differing only on the value of α:

   Consider a single neuron using the logistic function where data have just two features, x_1 and x_2. It thus has three weights, w_0, w_1, and w_2. Imagine that at the start of training all three weights are set to 0, that the first example you give has x_1=1 and x_2=1 and its label is +1, and that α= <see table>. You then do one update of the weights using that example using the usual update rule for logistic regression. After the update what output does the neuron give on a new example for which x_1=0 and x_2=0? (Please use three significant digits after the decimal. That means 12.99949 is 12.999 and -0.01251 is -0.013.)

   Because all three initial weights are zero it doesn't matter what the x_1 values are, $\overline{w}\cdot\overline{x} = 0$, so the activation is $g(\overline{w}\cdot\overline{x}) = \frac{1}{2}$. The update rule for weight w_j adds $\alpha x_j g'(\overline{w}\cdot\overline{x})(y - g(\overline{w}\cdot\overline{x}))$ to the weight. Since all the weights are set to zero that means each weight's new value is $\alpha x_j g'(\overline{w}\cdot\overline{x})\left(y - g(\overline{w}\cdot\overline{x})\right) = \alpha x_j g(\overline{w}\cdot\overline{x})(1 - g(\overline{w}\cdot\overline{x}))(y - g(\overline{w}\cdot\overline{x}))$. Plugging in the values for x_j (each of which equals 1) and $g(\overline{w}\cdot\overline{x})$ (which equals ½) gives us $\frac{\alpha}{8}$ as the new value for each of the weights.

   For the new example only x_0=1 is nonzero, so $\overline{w}\cdot\overline{x} = w_0 = \frac{\alpha}{8}$. This means $g(\overline{w}\cdot\overline{x}) = g\left(\frac{\alpha}{8}\right) = \frac{1}{1+e^{-\frac{\alpha}{8}}}$. This value is given in the table that follows for the different values of α used for the question.

α	$g\left(\frac{\alpha}{8}\right)$
0.5	a_i = 0.516
0.6	a_i=0.519
0.7	a_i=0.522
0.9	a_i=0.538

2. There was just one question, with different versions varying based on the values of the three weights and which weight, w1 or w2, is being asked for:

   The ReLU activation computes the function g(z) = z if z ≥ 0, 0 otherwise. Imagine we have a single neuron using this activation function that has just two inputs, x_1 and x_2 and thus three weights w_0, w_1, and w_2. The initial weights are set to w_0=<see below>, w_1=<see below>, and w_2=<see below> and we do one round of weight updates for an example for which x_1=x_2=1 and whose label is 0. α=1.0. What is the value of <see below> after the update?

   Since all the weights are negative and all the inputs are equal to 1, $\overline{w}\cdot\overline{x} < 0$, $g(\overline{w}\cdot\overline{x})$ outputs a 0. Note also that since $g(\overline{w}\cdot\overline{x})$ is constant in that area, $g'(\overline{w}\cdot\overline{x}) = 0$. The update rule adds $\alpha x_j g'(\overline{w}\cdot\overline{x})(y - g(\overline{w}\cdot\overline{x}))$ to each weight, but since $g'(\overline{w}\cdot\overline{x}) = 0$ it makes the entire weight update equal to 0, so none of the weights change in value. The answer is that the weight's value is the same value after the update as it was before the update.

w_0	w_1	w_2	Answer
-0.3	-0.2	-0.1	w_2=-0.1
-0.1	-0.2	-0.3	w_1=-0.2
-0.2	-0.1	-0.3	w_2=-0.3

3. There was just one question, differing in terms of the two weight update amounts:

Consider a node *j* within a multi-layer neural network. The node could be anywhere from the layer that the inputs directly feed into all the way to the top output layer. This question focuses on the bias $w_{0,j}$ of node *j*, and the weight $w_{i,j}$ of a link coming from node *i* and feeding into node *j*.

Imagine that after processing an example $w_{0,j}$ increased by <x: see below> and $w_{i,j}$ increased by <y: see below>. What was the activation a_i of neuron i? Please write your answer in decimal form with three digits of significance after the decimal.

(You may be perplexed that the question does not give you the specific activation function used nor the learning rate α. That is not an oversight.)

For a multi-layer network the weight update is $\alpha a_i \Delta_j$, so the amount $w_{0,j}$ is updated by is $\alpha a_0 \Delta_j$. I'll call the update amount for $w_{0,j}$ given in the question *x*. Since this weight is a bias, $a_0 = 1$ so we know that $\alpha a_0 \Delta_j = \alpha \Delta_j = x$. I'll call the update amount for $w_{i,j}$ given in the question *y*. We know that the formula for this update is $\alpha a_i \Delta_j$ and that it's equal to *y*, but we just figured out using the bias term that $\alpha \Delta_j = x$. Plugging that into the update equation tells us that $\alpha a_i \Delta_j = x a_i = y$, which means $a_i = \frac{y}{x}$. Your answer depended on the values of the two weight updates given in the question:

w_{0j}	w_{ij}	Answer
0.6	0.3	a_i=0.5
0.8	0.2	a_i=0.25
0.8	0.4	a_i=0.5
0.4	0.1	a_i=0.25
0.5	0.3	a_i=0.6