CS 4700: Foundations of Artificial Intelligence
Spring 2020
Prelim Prep Questions
Quiz 6

1. True/False: You can't apply neural network learning to data that are not linearly separable.

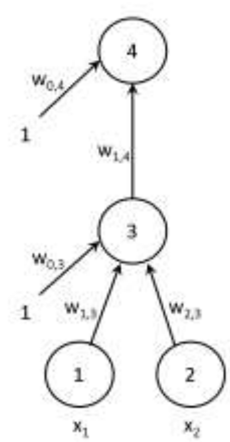
2. Image you define an activation function for a neuron as follows: $h_{\overline{\mathrm{w}}\cdot}(x) = 1$ if $\frac{1}{1+e^{-\mathrm{w}\cdot\mathrm{x}}} \geq 0.5$, and 0 otherwise. True or False: This will give the same output as a perceptron. If true, prove that this is so, otherwise give a counter-example.

3. True/False. When updating the weights of a multi-layer neural network using the backprop learning rule on a particular example, if the weight on a link coming out of a node is left unchanged, then none of the weights leading into that node will be changed. Explain your answer.

4. True/False. If an example is fed into the backprop learning rule and the current weights give the correct output, then backprop will leave the weights unchanged regardless of how many hidden layers the network has. If true, explain why. If false, give a counterexample.

5. Consider a logistic regression problem where data have two features x_1 and x_2, and where the two weights w_0 and w_1 are $-\ln(6)$ and $\ln(3)$ respectively (where ln represents natural logarithm). What must w_2 be in order for the logistic function to return the value 1/11 on an input with $x_1 = x_2 = 1$?

6. Consider the following two-node neural network:



For the purpose of this question nodes 3 and 4 use the logistic function as their activation function, and all of the weights are set to 1.0. In your answers do not compute the decimal representation of the value, keep it in symbolic form in terms of e.

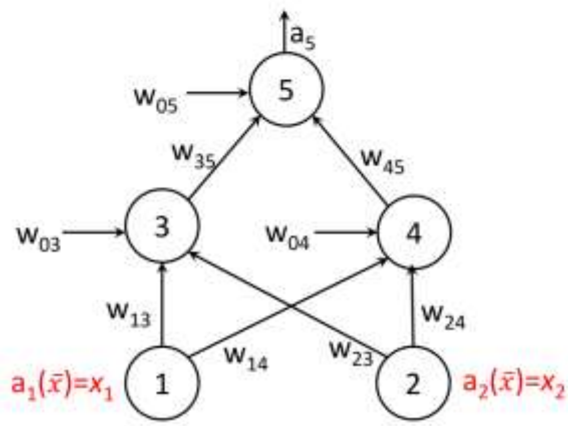
   a. Given the example whose two inputs are x_1=x_2=0.0, what is the activation of node 4?

b. If the desired output for the example in part a is 0 and you run it through backprop, what is Δ_4?

7. Consider using backprop with the activation function g(z) = z.

   a. What do the assignment statements in the backprop algorithm simplify to?

   b. Consider the following multi-layer neural network over two attributes given in the following diagram.



   Is it possible to express functions of x_1 and x_2 using this network that can't be equivalently expressed as a single neuron using this same activation function? If yes, give an example. If not, prove it.

1. True/False: You can't apply neural network learning to data that are not linearly separable.

   False. The algorithms will still apply. A perceptron will end up cycling through weights but you could still stop it at any point and take the weights it gets, such as they are. More importantly, the whole point of logistic regression and multi-layer backprop is so that they apply to data that aren't linearly separable.

2. Image you define an activation function for a neuron as follows: $h_{\overline{\mathrm{w}}\cdot}(x) = 1$ if $\frac{1}{1+e^{-\overline{\mathrm{w}}\cdot\overline{\mathrm{x}}}} \geq 0.5$, and 0 otherwise. True or False: This will give the same output as a perceptron. If true, prove that this is so, otherwise give a counter-example.

   True.

   $$\overline{w}\cdot\overline{x} \geq 0 \Leftrightarrow -(\overline{w}\cdot\overline{x}) \leq 0 \Leftrightarrow e^{-w\cdot x} \leq 1 \Leftrightarrow 1 + e^{-\overline{w}\cdot\overline{x}} \leq 2 \Leftrightarrow \frac{1}{1+e^{-\overline{w}\cdot\overline{x}}} \geq \frac{1}{2}$$

   Similarly,

   $$\overline{w}\cdot\overline{x} < 0 \Leftrightarrow -(\overline{w}\cdot\overline{x}) > 0 \Leftrightarrow e^{-\overline{w}\cdot\overline{x}} > 1 \Leftrightarrow 1 + e^{-\overline{w}\cdot\overline{x}} > 2 \Leftrightarrow \frac{1}{1+e^{-\overline{w}\cdot\overline{x}}} < \frac{1}{2}$$

   In other words they will both give the same output for the same $\overline{\mathrm{x}}$.

3. True/False. When updating the weights of a multi-layer neural network using the backprop learning rule on a particular example, if the weight on a link coming out of a node is left unchanged, then none of the weights leading into that node will be changed. Explain your answer.

   False. There are multiple reasons. For example, if there are multiple links coming out of the node but only one of them is zero and the others are non-zero, then they will also contribute to the amount that you change the weight coming in by.

4. True/False. If an example is fed into the backprop learning rule and the current weights give the correct output, then backprop will leave the weights unchanged regardless of how many hidden layers the network has. If true, explain why. If false, give a counterexample.

   True. Δ_j for all output nodes *j* would be zero, and all the other Δ_i would in turn be zero since you end up just adding a bunch of zeroes together.

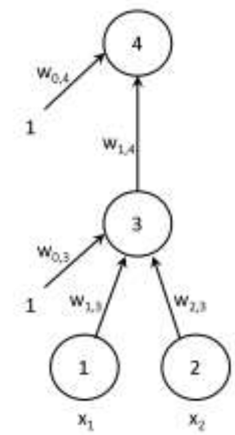
5. Consider a logistic regression problem where data have two features x_1 and x_2, and where the two weights w_0 and w_1 are –ln(6) and ln(3) respectively (where ln represents natural logarithm). What must w_2 be in order for the logistic function to return the value 1/11 on an input with $x_1 = x_2 = 1$?

   This is saying

   $$\frac{1}{1+e^{-(-\ln(6)+\ln(2)+w0)}} = \frac{1}{11}$$

This simplifies to w_0-ln(5) = ln(1/5) = ln(0.2).

6. Consider the following two-node neural network:



For the purpose of this question nodes 3 and 4 use the logistic function as their activation function, and all of the weights are set to 1.0. In your answers do not compute the decimal representation of the value, keep it in symbolic form in terms of e.

   a. Given the example whose two inputs are x_1=x_2=0.0, what is the activation of node 4?

$$a_3(\bar{x}) = \frac{1}{1+e^{-1}} = \frac{e}{e+1}$$

$$a_4(\bar{x}) = \frac{1}{1+e^{-(1+a_3(\bar{x}))}} = \frac{1}{1+e^{-\left(1+\frac{e}{e+1}\right)}} = \frac{e^{\left(\frac{2e+1}{e+1}\right)}}{e^{\left(\frac{2e+1}{e+1}\right)}+1}$$

   b. If the desired output for the example in part a is 0 and you run it through backprop, what is Δ_4?

$$\Delta_4(\bar{x}) = a_4(\bar{x})\big(1-a_4(\bar{x})\big)\big(0-a_4(\bar{x})\big) = a_4^3(\bar{x}) - a_4^2(\bar{x})$$
$$= \left(\frac{e^{\left(\frac{2e+1}{e+1}\right)}}{e^{\left(\frac{2e+1}{e+1}\right)}+1}\right)^3 - \left(\frac{e^{\left(\frac{2e+1}{e+1}\right)}}{e^{\left(\frac{2e+1}{e+1}\right)}+1}\right)^2$$
$$= \frac{\left(e^{\left(\frac{2e+1}{e+1}\right)}\right)^2\left(e^{\left(\frac{2e+1}{e+1}\right)} - \left(e^{\left(\frac{2e+1}{e+1}\right)}+1\right)\right)}{\left(e^{\left(\frac{2e+1}{e+1}\right)}+1\right)^3}$$

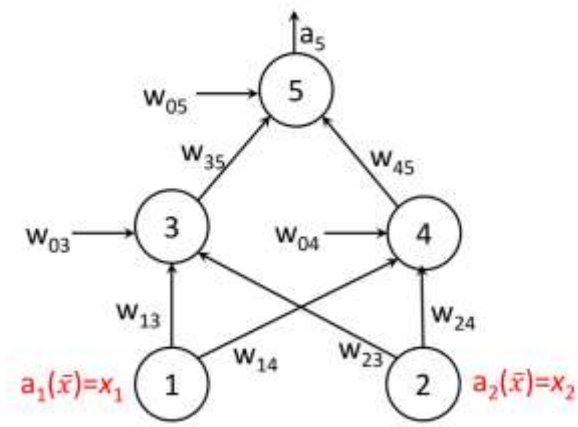
$$= -\frac{e^{\left(\frac{2e+1}{e+1}\right)^2}}{\left(e^{\left(\frac{2e+1}{e+1}\right)} + 1\right)^3}$$

7. Consider using backprop with the activation function g(z) = z.

   a. What do the assignment statements in the backprop algorithm simplify to?

   It is the assignment where $g'(\overline{w} \cdot \bar{x})$ is dropped, since the first derivative is just 1.

   b. Consider the following multi-layer neural network over two attributes given in the following diagram.



Is it possible to express functions of x_1 and x_2 using this network that can't be equivalently expressed as a single neuron using this same activation function? If yes, give an example. If not, prove it.

No. Linear functions are closed under composition, meaning a linear sum of linear sums is itself a linear sum:

$g_3(x) = w_{03} + w_{13}x_1 + w_{23}x_2$

$g_4(x) = w_{04} + w_{14}x_1 + w_{24}x_2$

$g_5(x) = w_{05} + w_{35}g_3(x) + w_{45}g_4(x)$

$= w_{05} + w_{35}\,(w_{03} + w_{13}x_1 + w_{23}x_2) + w_{45}\,(w_{04} + w_{14}x_1 + w_{24}x_2)$

$= (w_{05} + w_{35}\,w_{03} + w_{45}\,w_{04}) + (w_{35}\,w_{13} + w_{45}\,w_{14})\,x_1 + (w_{35}\,w_{23} + w_{45}\,w_{24})\,x_2$

Thus any such network with the given weights is equivalent to a single-layer network whose weights are $w_0 = (w_{05} + w_{35} w_{03} + w_{45} w_{04})$, $w_1 = (w_{35} w_{13} + w_{45} w_{14})$, and $w_2 = (w_{35} w_{23} + w_{45} w_{24})$.