DSC 140B - Quiz 06

February 19, 2026

Name:

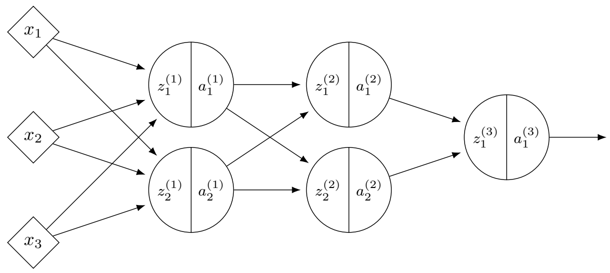
PID:

About the quizzes:

- Quizzes in DSC 140B are *optional* and graded pass/fail.
- A score of 70% or higher earns a "pass" and 1.5 credits toward your final grade.
- If you don't pass, no credits are earned, but it doesn't hurt your grade.
- You have 30 minutes to complete the quiz.
- At least one of the questions below will be on an exam (probably with slight changes, such as different numbers).
- Unfortunately, we can't answer clarifying questions during the quiz. If you think a question has a bug or is unclear, please let us known in a private post on Campuswire after the quiz, and we'll take it into account when grading.

Problem 1.

Consider the neural network $H(\vec{x})$ shown below:



Let the weights of the network be:

$$W^{(1)} = \begin{pmatrix} 2 & 1 \\ -1 & 3 \\ 1 & -2 \end{pmatrix} \quad W^{(2)} = \begin{pmatrix} 4 & -1 \\ 2 & 1 \end{pmatrix} \quad W^{(3)} = \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

Assume that all nodes use **linear** activation functions, and that all biases are zero.

Suppose $\vec{x} = (1, -1, 2)^T$.

a) ($\frac{1}{2}$ point) What is $a_1^{(1)}$?

5

Solution: 5.

With linear activations, $a = z$ at every node.

We find:

$$\begin{aligned} z_1^{(1)} &= W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 + W_{31}^{(1)} x_3 \\ &= 2(1) + (-1)(-1) + 1(2) \\ &= 5 \end{aligned}$$

This was similar to Practice Problem 125.

b) ($\frac{1}{2}$ point) What is $a_2^{(2)}$?

-11

Solution: -11.

First, we need $\vec{a}^{(1)}$. From part (a), $a_1^{(1)} = 5$. We also need:

$$\begin{aligned} z_2^{(1)} &= W_{12}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{32}^{(1)} x_3 \\ &= 1(1) + 3(-1) + (-2)(2) \\ &= -6 \end{aligned}$$

So $\vec{a}^{(1)} = (5, -6)^T$. Now:

$$\begin{aligned} z_2^{(2)} &= W_{12}^{(2)} a_1^{(1)} + W_{22}^{(2)} a_2^{(1)} \\ &= -1(5) + 1(-6) \\ &= -11 \end{aligned}$$

This was similar to Practice Problem 125.

c) What is $H(\vec{x})$?

46

Solution: 46.

We know that $H(\vec{x}) = z_1^{(3)} = W_{11}^{(3)} a_1^{(2)} + W_{21}^{(3)} a_2^{(2)}$. We already know $a_2^{(2)}$ from part (b). We need $a_1^{(2)}$:

$$\begin{aligned} z_1^{(2)} &= W_{11}^{(2)} a_1^{(1)} + W_{21}^{(2)} a_2^{(1)} \\ &= 4(5) + 2(-6) \\ &= 8 \end{aligned}$$

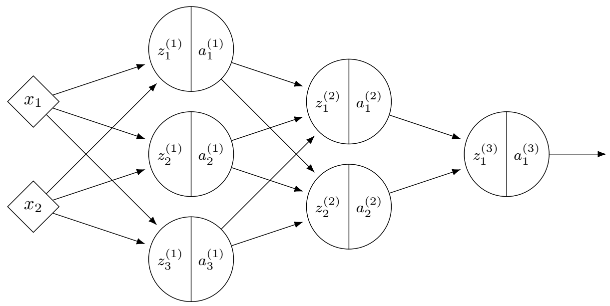
So $\vec{a}^{(2)} = (8, -11)^T$. Now:

$$\begin{aligned} H(\vec{x}) &= W_{11}^{(3)} a_1^{(2)} + W_{21}^{(3)} a_2^{(2)} \\ &= 3(8) + (-2)(-11) \\ &= 46 \end{aligned}$$

This was similar to Practice Problem 125.

Problem 2.

Consider the neural network $H(\vec{x})$ shown below:



Let the weights of the network be:

$$W^{(1)} = \begin{pmatrix} 1 & -2 & 3 \\ 2 & 1 & -1 \end{pmatrix} \quad W^{(2)} = \begin{pmatrix} 2 & -1 \\ 1 & 3 \\ -2 & 1 \end{pmatrix} \quad W^{(3)} = \begin{pmatrix} 4 \\ -3 \end{pmatrix}$$

Assume that all hidden nodes use **ReLU activation** functions, that the output node uses a linear activation, and that all biases are zero.

Suppose $\vec{x} = (1, 3)^T$.

a) ($\frac{1}{2}$ point) What is $a_3^{(1)}$?

0

Solution: 0.

With ReLU activations, $a = \text{ReLU}(z)$ at every hidden node.

We find:

$$\begin{aligned} z_3^{(1)} &= W_{13}^{(1)} x_1 + W_{23}^{(1)} x_2 \\ &= 3(1) + (-1)(3) \\ &= 0 \end{aligned}$$

So $a_3^{(1)} = \text{ReLU}(0) = 0$.

This was similar to Practice Problem 126.

b) ($\frac{1}{2}$ point) What is $a_1^{(2)}$?

15

Solution: 15.

First, we need $\vec{a}^{(1)}$. From part (a), $a_3^{(1)} = 0$. We also need:

$$\begin{aligned} z_1^{(1)} &= W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 \\ &= 1(1) + 2(3) \\ &= 7 \end{aligned}$$

So $a_1^{(1)} = \text{ReLU}(7) = 7$.

$$\begin{aligned} z_2^{(1)} &= W_{12}^{(1)} x_1 + W_{22}^{(1)} x_2 \\ &= -2(1) + 1(3) \\ &= 1 \end{aligned}$$

So $a_2^{(1)} = \text{ReLU}(1) = 1$. Thus $\vec{a}^{(1)} = (7, 1, 0)^T$. Now:

$$\begin{aligned} z_1^{(2)} &= W_{11}^{(2)} a_1^{(1)} + W_{21}^{(2)} a_2^{(1)} + W_{31}^{(2)} a_3^{(1)} \\ &= 2(7) + 1(1) + (-2)(0) \\ &= 15 \end{aligned}$$

So $a_1^{(2)} = \text{ReLU}(15) = 15$.

This was similar to Practice Problem 126.

c) What is $H(\vec{x})$?

60

Solution: 60.

We know that $H(\vec{x}) = z_1^{(3)} = W_{11}^{(3)} a_1^{(2)} + W_{21}^{(3)} a_2^{(2)}$. We already know $a_1^{(2)}$ from part (b). We need $a_2^{(2)}$:

$$\begin{aligned} z_2^{(2)} &= W_{12}^{(2)} a_1^{(1)} + W_{22}^{(2)} a_2^{(1)} + W_{32}^{(2)} a_3^{(1)} \\ &= -1(7) + 3(1) + 1(0) \\ &= -4 \end{aligned}$$

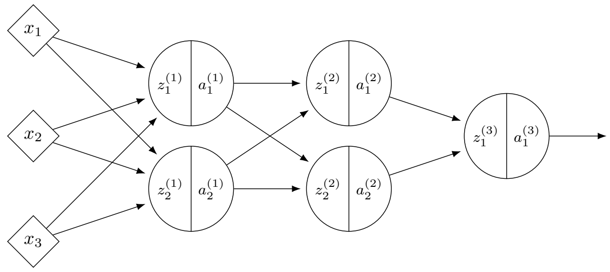
So $a_2^{(2)} = \text{ReLU}(-4) = 0$. Thus $\vec{a}^{(2)} = (15, 0)^T$. Now:

$$\begin{aligned} H(\vec{x}) &= W_{11}^{(3)} a_1^{(2)} + W_{21}^{(3)} a_2^{(2)} \\ &= 4(15) + (-3)(0) \\ &= 60 \end{aligned}$$

This was similar to Practice Problem 126.

Problem 3.

Consider the neural network $H(\vec{x})$ shown below:



The first layer of this neural network can be thought of as a function $f : \mathbb{R}^3 \to \mathbb{R}^k$ mapping the input feature vector in $\mathbb{R}^3$ to a new representation. Assume all activations are **linear** and all biases are zero.

What is this new representation if

$$W^{(1)} = \begin{pmatrix} 3 & -1 \\ 1 & 2 \\ -2 & 1 \end{pmatrix}$$

and $\vec{x} = (2, 1, -1)^T$?

- ○ $(-1, 9)^T$
- ○ $(9, -1, 0)^T$
- ● $(9, -1)^T$
- ○ $(5, 3, -1)^T$

Solution: $(9, -1)^T$.

The new representation is $(z_1^{(1)}, z_2^{(1)})^T$, so we need to compute $z_1^{(1)}$ and $z_2^{(1)}$:

$$z_1^{(1)} = 3(2) + 1(1) + (-2)(-1) = 6 + 1 + 2 = 9$$
$$z_2^{(1)} = -1(2) + 2(1) + 1(-1) = -2 + 2 - 1 = -1$$

So the new representation is $(9, -1)^T$.

This was similar to Practice Problem 128.

Problem 4.

Suppose H is a neural network with the following architecture: 2 input features, a first hidden layer with 3 nodes, a second hidden layer with 3 nodes, and 1 output node. Every hidden and output node has a bias. You may assume that the network is fully connected, and feed-forward.

The gradient of H with respect to the parameters is a vector. What is this vector's dimensionality?

- ○ 18
- ○ 22
- ● 25
- ○ 28

Solution: 25.

We need to count all weights and biases. You can do this by drawing out the network and counting, but you will quickly discover a pattern: the number of weights between two layers is the product of the number of nodes in those layers, and the number of biases in a layer is the number of nodes in that layer. So, we have:

Layer 1: 2×3 weights + 3 biases = 9.

Layer 2: 3×3 weights + 3 biases = 12.

Output: 3×1 weights + 1 bias = 4.

Total: $9 + 12 + 4 = 25$.

This was similar to Practice Problem 129.

Problem 5.

You are predicting salaries (in dollars) with a neural network.

True or False: sigmoid activation is a good choice for the output node.

○ True

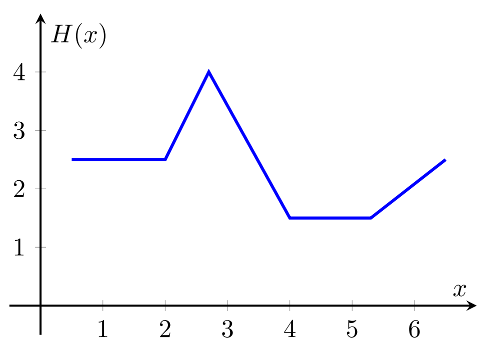
● False

Solution: False.

The sigmoid function outputs values between 0 and 1. Salaries can be much larger than 1 (e.g., $50,000), so sigmoid activation at the output cannot represent typical salary values. A linear activation would be more appropriate for the output node in a regression task like this.

Problem 6.

Suppose that when a deep neural network $H(x) : \mathbb{R} \to \mathbb{R}$ is plotted, the resulting graph looks like the following:



True or False: it is possible that the network uses **ReLU** activation in its hidden layers, and **ReLU** activation in its output layer.

● True

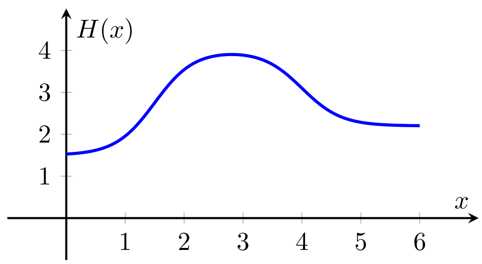
○ False

Solution: True. Applying ReLU at the output clips negative values to zero. A deep ReLU network with linear output produces a piecewise linear function. If the piecewise linear function produced by the hidden layers already has the shape shown (non-negative everywhere), the output ReLU would not change it. The plot is consistent with this.

This was Practice Problem 132 (b).

Problem 7.

Suppose that when a deep neural network $H(x) : \mathbb{R} \to \mathbb{R}$ is plotted, the resulting graph looks like the following:



a) True or False: it is possible that the network uses **sigmoid** activations in its hidden layers and a **linear** activation in its output layer.

- ● True
- ○ False

Solution: True.

Sigmoid activations in the hidden layers produce smooth, curved outputs. A linear output node computes a weighted sum of these smooth functions, which can produce a smooth curve with multiple bends, like the one shown. In fact, the plot above was generated by a sum of sigmoid functions, which is exactly what such a network computes.

b) True or False: it is possible that the network uses **linear** activations in its hidden layers and a **sigmoid** activation in its output layer.

- ○ True
- ● False

Solution: False.

We said in lecture that the composition of linear (affine) functions is itself a linear (affine) function. No matter how many hidden layers the network has, if they all use linear activations, the entire network before the output reduces to a single affine function $f(x) = wx + b$.

In other words, if the network had a linear activation function on the output node, the plot would be a single straight line. If we use a sigmoid activation instead, then we feed that straight line into the sigmoid function. The result is just a sigmoid curve, which is a single bend. You can see this intuitively, or you can verify it mathematically by plugging $f(x) = wx + b$ into the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$, getting:

$$\sigma(wx + b) = \frac{1}{1 + e^{-(wx+b)}}.$$

This function has the same form as the standard sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$, except that the input z is replaced by a linear function of x. The parameters w and b control the steepness and horizontal shift of the sigmoid curve, but they do not change its fundamental shape.

Therefore, the output of a network with linear activations in the hidden layers and a sigmoid activation in the output layer is a sigmoid function of a linear function, which is always a single monotonic sigmoid curve. The multi-bend curve shown above is not possible.