Congratulations! You passed!

Grade received 80%   Latest Submission Grade 80%   To pass 80% or higher

Go to next item

1. Which of the following are true? (Check all that apply.) **1 / 1 point**

- [ ] $a_3^{[2]}$ denotes the activation vector of the second layer for the third example.
- [x] $a^{[2]}$ denotes the activation vector of the second layer.

  ✓ **Correct**
  Yes. In our convention $a^{[j]}$ denotes the activation function of the j-th layer.

- [x] $w_3^{[4]}$ is the column vector of parameters of the fourth layer and third neuron.

  ✓ **Correct**
  Yes. The vector $w_j^{[i]}$ is the column vector of parameters of the i-th layer and j-th neuron of that layer.

- [ ] $w_3^{[4]}$ is the row vector of parameters of the fourth layer and third neuron.
- [ ] $w_3^{[4]}$ is the column vector of parameters of the third layer and fourth neuron.
- [ ] $a^{[3](2)}$ denotes the activation vector of the second layer for the third example.

Expand

Correct
Great, you got all the right answers.

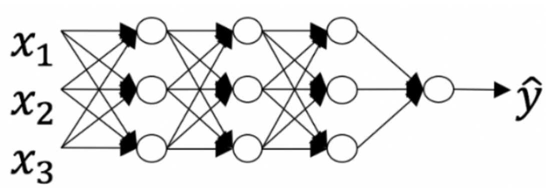
2. In which of the following cases is the linear (identity) activation function most likely used? **1 / 1 point**

- ( ) As activation function in the hidden layers.
- (•) When working with regression problems.
- ( ) For binary classification problems.
- ( ) The linear activation function is never used.

Expand

Correct
Yes. In problems such as predicting the price of a house it makes sense to use the linear activation function as output.

3. Which of the following represents the activation output of the second neuron of the third layer applied to the fourth example? **1 / 1 point**



- (•) $a_2^{[3](4)}$
- ( ) $a_2^{[4](3)}$
- ( ) $a_4^{[3](2)}$
- ( ) $a_3^{[4]2}$

Expand

Correct
Yes. The superscript in brackets indicates the layer number, the superscript in parenthesis represents the number of examples, and the subscript the number of the neuron.

4. You are building a binary classifier for recognizing cucumbers (y=1) vs. watermelons (y=0). Which one of these activation functions would you recommend using for the output layer?

1 / 1 point

- (selected) sigmoid
- Leaky ReLU
- tanh
- ReLU

Expand

Correct
Yes. Sigmoid outputs a value between 0 and 1 which makes it a very good choice for binary classification. You can classify as 0 if the output is less than 0.5 and classify as 1 if the output is more than 0.5. It can be done with tanh as well but it is less convenient as the output is between -1 and 1.

5. Consider the following code:

1 / 1 point

#+begin_src python

x = np.random.rand(3, 2)

y = np.sum(x, axis=0, keepdims=True)

#+end_src

What will be y.shape?

- (3, 1)
- (3,)
- (selected) (1, 2)
- (2,)

Expand

Correct
Yes. By choosing the axis=0 the sum is computed over each column of the array, thus the resulting array is a row vector with 2 entries. Since the option keepdims=True is used the first dimension is kept, thus (1, 2).

6. Suppose you have built a neural network. You decide to initialize the weights and biases to be zero. Which of the following statements is true?

1 / 1 point

- (selected) Each neuron in the first hidden layer will perform the same computation. So even after multiple iterations of gradient descent, each neuron in the layer will be computing the same thing as other neurons.
- Each neuron in the first hidden layer will compute the same thing, but neurons in different layers will compute different things, thus we have accomplished "symmetry breaking" as described in the lecture.
- The first hidden layer's neurons will perform different computations from each other even in the first iteration; their parameters will thus keep evolving in their own way.
- Each neuron in the first hidden layer will perform the same computation in the first iteration. But after one iteration of gradient descent they will learn to compute different things because we have "broken symmetry".

Expand

Correct

7. Logistic regression's weights should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to "break symmetry", True/False?

1 / 1 point

- True
- (selected) False

Expand

Correct

Yes, Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed into the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights' values follow x's distribution and are different from each other if x is not a constant vector.

8. Which of the following are true about the tanh function?

0 / 1 point

- [ ] For large values the slope is larger.
- [ ] For large values the slope is close to zero.
- [x] The tanh is mathematically a shifted version of the sigmoid function.

Correct

Yes. You can see the shape of both is very similar but tanh passes through the origin.

- [ ] The slope is zero for negative values.
- [x] The derivative at $c = 0$ is not well defined.

! **This should not be selected**

No. The slope of the tangent line of the tanh function is well defined at 0, moreover the slope is 1.

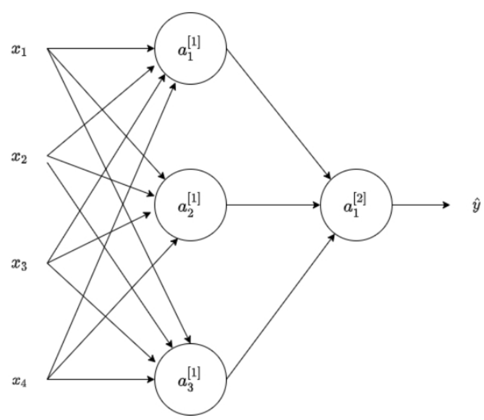
Expand

Incorrect

You didn't select all the correct answers

9. Consider the following 1 hidden layer neural network:

1 / 1 point



Which of the following statements are True? (Check all that apply).

- [x] $W^{[1]}$ will have shape (3, 4).

Correct

Yes. The number of rows in $W^{[k]}$ is the number of neurons in the k-th layer and the number of columns is the number of inputs of the layer.

- [x] $b^{[1]}$ will have shape (3, 1).

Correct

Yes. $b^{[k]}$ is a column vector and has the same number of rows as neurons in the k-th layer.

- [ ] $b^{[1]}$ will have shape (1, 3)
- [ ] $b^{[2]}$ will have shape (3, 1)
- [x] *E: $b^{[2]}$ will have shape (1,1)

Correct

Yes. $b^{[k]}$ is a column vector and has the same number of rows as neurons in the k-th layer.

- [ ] $W^{[1]}$ will have shape (4, 3).

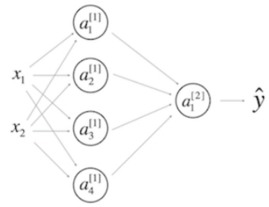
Expand

Correct

Great, you got all the right answers.

10. What are the dimensions of $Z^{[1]}$ and $A^{[1]}$?

0 / 1 point



- ( ) $Z^{[1]}$ and $A^{[1]}$ are (4,m)
- (•) $Z^{[1]}$ and $A^{[1]}$ are (4,1)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (1,4)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (4,2)

Expand

Incorrect

Remember that $Z^{[1]}$ and $A^{[1]}$ are quantities computed over a batch of training examples, not only 1.