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.)

0 / 1 point

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

- [ ] $w_3^{[4]}$ 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.
- [ ] $a^{[3](2)}$ denotes the activation vector of the second layer for the third example.

Expand

Incorrect
You didn't select all the correct answers

2. The tanh activation is not always better than sigmoid activation function for hidden units because the mean of its output is closer to zero, and so it centers the data, making learning complex for the next layer. True/False?

1 / 1 point

- ( ) True
- (•) False

Expand

Correct
Yes. As seen in lecture the output of the tanh is between -1 and 1, it thus centers the data which makes the learning simpler for the next layer.

3. Which of these is a correct vectorized implementation of forward propagation for layer l, where $1 \leq l \leq L$?

1 / 1 point

- (•) $Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$
  $A^{[l]} = g^{[l]}(Z^{[l]})$
- ( ) $Z^{[l]} = W^{[l]}A^{[l]} + b^{[l]}$
  $A^{[l+1]} = g^{[l]}(Z^{[l]})$
- ( ) `$$Z^{[l]} = W^{[l-1]} A^{[l]}+ b^{[l-1]}$$`
  `$$A^{[l]} = g^{[l]}(Z^{[l]})$$`
- ( ) $Z^{[l]} = W^{[l]}A^{[l]} + b^{[l]}$

Expand

Correct

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

- ( ) Leaky ReLU
- (•) sigmoid
- ( ) 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?

- (1, 2)
- (3,)
- (3, 1)
- (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?

0 / 1 point

- 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.
- 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.
- 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

Incorrect
You did not choose an option.

7. Using linear activation functions in the hidden layers of a multilayer neural network is equivalent to using a single layer. True/False?

1 / 1 point

- True
- False

Expand

Correct
Yes. When the identity or linear activation function $g(c) = c$ is used the output of composition of layers is equivalent to the computations made by a single layer.

8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relatively large values, using np.random.randn(..,..)*1000. What will happen?

1 / 1 point

- So long as you initialize the weights randomly gradient descent is not affected by whether the weights are large or small.
- This will cause the inputs of the tanh to also be very large, causing the units to be "highly activated" and thus speed up learning compared to if the weights had to start from small values.
- This will cause the inputs of the tanh to also be very large, thus causing gradients to also become large. You therefore have to set

  α

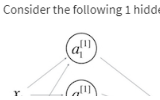
  to a very small value to prevent divergence; this will slow down learning.
- This will cause the inputs of the tanh to also be very large, thus causing gradients to be

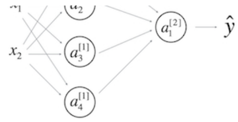
Expand

Correct
Yes. tanh becomes flat for large values; this leads its gradient to be close to zero. This slows down the optimization algorithm.

9. Consider the following 1 hidden layer neural network:

1 / 1 point





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

- [ ] $b^{[1]}$ will have shape (2, 1)
- [ ] $$W^{[2]}$$ will have shape (4, 1)
- [x] $$b^{[2]}$$ will have shape (1, 1)

Correct

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

Correct

- [x] $$W^{[2]}$$ will have shape (1, 4)

Correct

- [ ] $$b^{[2]}$$ will have shape (4, 1)
- [x] $$W^{[1]}$$ will have shape (4, 2)

Correct

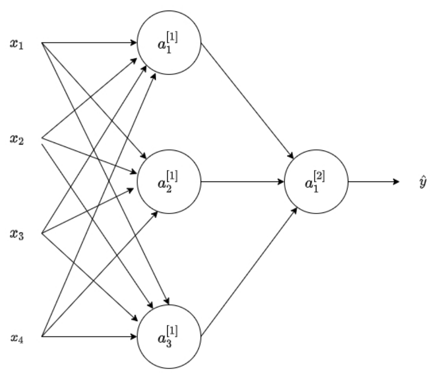
- [ ] $$W^{[1]}$$ will have shape (2, 4)

Expand

Correct
Great, you got all the right answers.

10. Consider the following 1 hidden layer neural network:

1 / 1 point



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

- ( ) $Z^{[1]}$ and $A^{[1]}$ are (4, m)
- (•) $Z^{[1]}$ and $A^{[1]}$ are (3, m)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (3, 1)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (4, 1)

Expand

Correct
Yes. The $Z^{[1]}$ and $A^{[1]}$ are calculated over a batch of training examples. The number of columns in $Z^{[1]}$ and $A^{[1]}$ is equal to the number of examples in the batch, m. And the number of rows in $Z^{[1]}$ and $A^{[1]}$ is equal to the number of neurons in the first layer.