Congratulations! You passed!

Grade received 90% **Latest Submission Grade** 90% **To pass** 80% or higher

Retake the assignment in **23h 27m**

Go to next item

1. What is the "cache" used for in our implementation of forward propagation and backward propagation?

1 / 1 point

- ○ We use it to pass variables computed during backward propagation to the corresponding forward propagation step. It contains useful values for forward propagation to compute activations.
- ○ It is used to keep track of the hyperparameters that we are searching over, to speed up computation.
- ○ It is used to cache the intermediate values of the cost function during training.
- ◉ We use it to pass Z computed during forward propagation to the corresponding backward propagation step. It contains useful values for backward propagation to compute derivatives.

Expand

Correct
Correct, the "cache" records values from the forward propagation units and are used in backward propagation units because it is needed to compute the chain rule derivatives.

2. During the backpropagation process, we use gradient descent to change the hyperparameters. True/False?

1 / 1 point

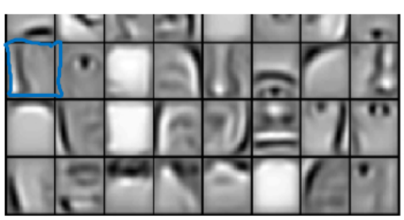
- ○ True
- ◉ False

Expand

Correct
Correct. During backpropagation, we use gradient descent to compute new values of $$W^{[l]}$$ and $$b^{[l]}$$. These are the parameters of the network.

3. Which of the following is more likely related to the early layers of a deep neural network?

1 / 1 point



Expand

Correct
Yes. The early layer of a neural network usually computes simple features such as edges and lines.

4. Vectorization allows you to compute forward propagation in an L-layer neural network without an explicit for-loop (or any other explicit iterative loop) over the layers l=1, 2, …,L. True/False?

1 / 1 point

- ○ True
- ◉ False

Expand

Correct
Forward propagation propagates the input through the layers, although for shallow networks we may just write all the lines ($$a^{[2]} = g^{[2]}(z^{[2]})$$, $$z^{[2]}= W^{[2]}a^{[1]}+b^{[2]}$$, ...) in a deeper network, we cannot avoid a for loop iterating over the layers: ($$a^{[l]} = g^{[l]}(z^{[l]})$$, $$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$$, ...).

5. Assume we store the values for $n^{[l]}$ in an array called layer_dims, as follows: layer_dims = [$n_x n$,4,3,2,1]. So layer 1 has four hidden units, layer 2 has 3 hidden units, and so on. Which of the following for-loops will allow you to initialize the parameters for the model?

0 / 1 point

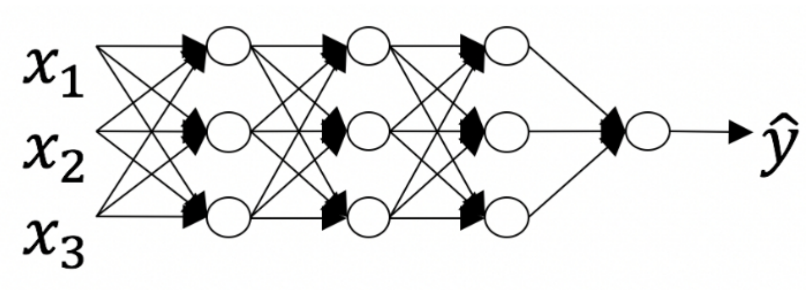
- ○
```
for i in range(1, len(layer_dims)/2):
parameter['W' + str(i)] = np.random.randn(layer_dims[i], layer_dims[i-1]) * 0.01
parameter['b' + str(i)] = np.random.randn(layer_dims[i], 1) * 0.01
```
- ○
```
for i in range(len(layer_dims)-1):
parameter['W' + str(i+1)] = np.random.randn(layer_dims[i+1], layer_dims[i]) * 0.01
parameter['b' + str(i+1)] = np.random.randn(layer_dims[i+1], 1) * 0.01
```
- ◉
```
for i in range(len(layer_dims)-1):
parameter['W' + str(i+1)] = np.random.randn(layer_dims[i], layer_dims[i+1]) * 0.01
parameter['b' + str(i+1)] = np.random.randn(layer_dims[i+1], 1) * 0.01
```
- ○
```
for i in range(len(layer_dims)):
parameter['W' + str(i+1)] = np.random.randn(layer_dims[i+1], layer_dims[i]) * 0.01
parameter['b' + str(i+1)] = np.random.randn(layer_dims[i+1], 1) * 0.01
```

Expand

Incorrect
No. This gives the $$W^{[l]}$$ the wrong shape ($$n^{[l]}$$, $$n^{[l+1]}$$).

6. Consider the following neural network.

1 / 1 point



How many layers does this network have?

- ○ The number of layers L is 5. The number of hidden layers is 4.
- ○ The number of layers L is 3. The number of hidden layers is 3.
- ○ The number of layers L is 4. The number of hidden layers is 4.
- ◉ The number of layers L is 4. The number of hidden layers is 3.

Expand

Correct
Yes. As seen in lecture, the number of layers is counted as the number of hidden layers + 1. The input and output layers are not counted as hidden layers.

7. During forward propagation, for the value of $A^{[l]}$ the value is used of $Z^{[l]}$ with the activation function $g^{[l]}$. During backward propagation we calculate $dA^{[l]}$ from $Z^{[l]}$.

1 / 1 point

- [ ] True
- [x] False

Expand

Correct

Correct. During backward propagation we are interested in computing $$dW^{[l]}$$ and $$db^{[l]}$$. For that we use $$g'^{L}$$, $$dZ^{[l]}$$, $$Z^{[l]}$$, and $$W^{[l]}$$.

8. There are certain functions with the following properties:

1 / 1 point

(i) To compute the function using a shallow network circuit, you will need a large network (where we measure size by the number of logic gates in the network), but (ii) To compute it using a deep network circuit, you need only an exponentially smaller network. True/False?

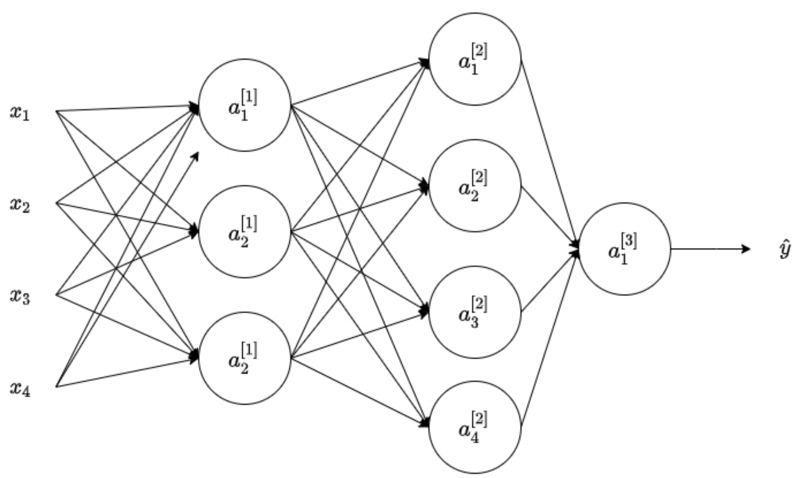
- [ ] False
- [x] True

Expand

Correct

9. Consider the following 2 hidden layers neural network:

1 / 1 point



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

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

Correct

Yes. More generally, the shape of $b^{[l]}$ is $(n^{[l]}, 1)$.

- [ ] $b^{[1]}$ will have shape (1, 3)
- [ ] $W^{[2]}$ will have shape (1, 3)
- [ ] $W^{[2]}$ will have shape (3, 1)

- [ ] $W^{[2]}$ will have shape (3, 4)
- [ ] $b^{[1]}$ will have shape (4, 1)
- [x] $W^{[1]}$ will have shape (3, 4)

Correct
Yes. More generally, the shape of $W^{[l]}$ is $(n^{[l]}, n^{[l-1]})$.

- [x] $W^{[2]}$ will have shape (4, 3)

Correct
Yes. More generally, the shape of $W^{[l]}$ is $(n^{[l]}, n^{[l-1]})$.

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

Expand

Correct
Great, you got all the right answers.

10. In the general case if we are training with m examples what is the shape of $A^{[l]}$?

1 / 1 point

- ( ) $(n^{[l+1]}, m)$
- ( ) $(m, n^{[l+1]})$
- (•) $(n^{[l]}, m)$
- ( ) $(m, n^{[l]})$

Expand

Correct
Yes. The number of rows in $$A^{[1]}$$ corresponds to the number of units in the l-th layer.