Congratulations! You passed!

Grade received 97.50% **To pass** 80% or higher

Go to next item

Key Concepts on Deep Neural Networks

Latest Submission Grade 97.5%

1. We use the "cache" in our implementation of forward and backward propagation to pass useful values to the next layer in the forward propagation. True/False? — 1 / 1 point

- ( ) False
- ( ) True

Correct. The "cache" is used in our implementation to store values computed during forward propagation to be used in backward propagation.

2. Which of the following are "parameters" of a neural network? (Check all that apply.) — 1 / 1 point

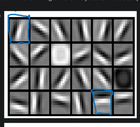
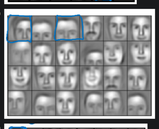
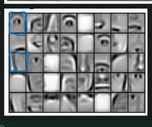
- [x] $b^{[l]}$ the bias vector.

Correct. The weight matrices and the bias vectors are the parameters of the network.

- [ ] $g^{[l]}$ the activation functions.
- [ ] L the number of layers of the neural network.
- [x] $W^{[l]}$ the weight matrices.

Correct. The weight matrices and the bias vectors 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

- (•) 
- ( ) 
- ( ) 

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

4. We can not use vectorization to calculate $dA^{[l]}$ in backpropagation, we must use a for loop over all the examples. True/False? — 1 / 1 point

- ( ) True
- (•) False

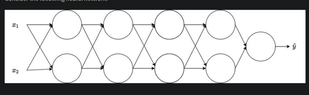
Correct. We can use vectorization in backpropagation to calculate $dA^{[l]}$ for each layer. This computation is done over all the training examples.

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? — 1 / 1 point

- (•)
```
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)):

    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(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], layer_dims[i+1]) * 0.01

    parameter['b' + str(i+1)] = np.random.randn(layer_dims[i+1], 1) * 0.01
```

Correct. Yes. This iterates over 0, 1, 2, 3 and assigns to $W^{[i]}$ the shape $(n^{[i]}, n^{[i-1]})$.

6. Consider the following neural network: — 1 / 1 point



How many layers does this network have?

- ( ) The number of layers L is 2.
- ( ) The number of layers L is 6
- ( ) The number of layers L is 4.
- (•) The number of layers L is 5.

Correct. Yes. The number of layers is the number of hidden layers + 1.

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

- (•) False
- ( ) True

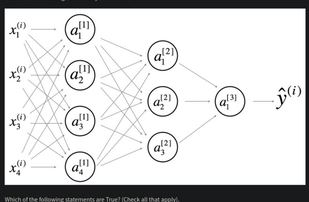
Correct. Correct. During backward propagation we are interested in computing $dW^{[l]}$ and $db^{[l]}$. For that we use $g^{[l]}$, $dZ^{[l]}$, $Z^{[l]}$, and $W^{[l]}$.

8. For any mathematical function you can compute with an L-layered deep neural network with N hidden units there is a shallow neural network that requires only $\log N$ units, but it is very difficult to train. — 1 / 1 point

- (•) False
- ( ) True

Correct. Correct. On the contrary, some mathematical functions can be computed using an L-layered neural network and a given number of hidden units; but using a shallow neural network the number of necessary hidden units grows exponentially.

9. Consider the following 2 hidden layer neural network: — 0.75 / 1 point



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

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

This should not be selected. No. More generally, the shape of $b^{[l]}$ is $(n^{[l]}, 1)$.

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

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

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

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

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

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

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

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

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

10. In the general case if we are training with m examples what is the shape of $A^{[l]}$? — 1 / 1 point

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

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