Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

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

- 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 variables computed during backward propagation to the corresponding forward propagation step. It contains useful values for forward propagation to compute activations.
- (selected) 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. Among the following, which ones are "hyperparameters"? (Check all that apply.) **1 / 1 point**

- [ ] weight matrices $W^{[l]}$
- [x] learning rate α

  ✓ Correct
- [x] number of layers L in the neural network

  ✓ Correct
- [x] size of the hidden layers $n^{[l]}$

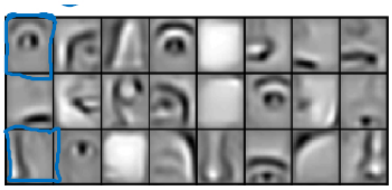
  ✓ Correct
- [x] number of iterations

  ✓ Correct
- [ ] bias vectors $b^{[l]}$
- [ ] activation values $a^{[l]}$

Expand

Correct
Great, you got all the right answers.

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. 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

- (•) False
- ( ) True

Expand

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

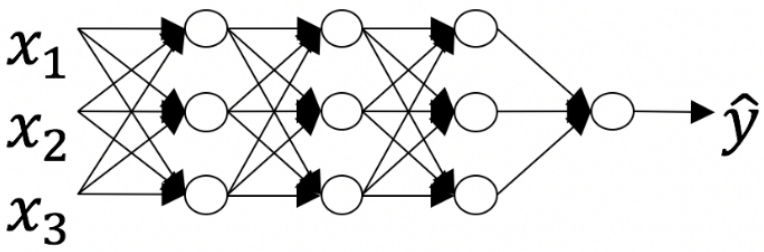
5. Assume we store the values for $n^{[l]}$ in an array called layer_dims, as follows: layer_dims = [n_x, 4,3,2,1]. So layer 1 has four hidden units, layer 2 has 3 hidden units and so on. Which of the following for-loops will allow you to initialize the parameters for the model? 1 / 1 point

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

Expand

Correct

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



How many layers does this network have?

- ( ) The number of layers L is 4. 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 3.
- ( ) The number of layers L is 5. The number of hidden layers is 4.

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
- 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**

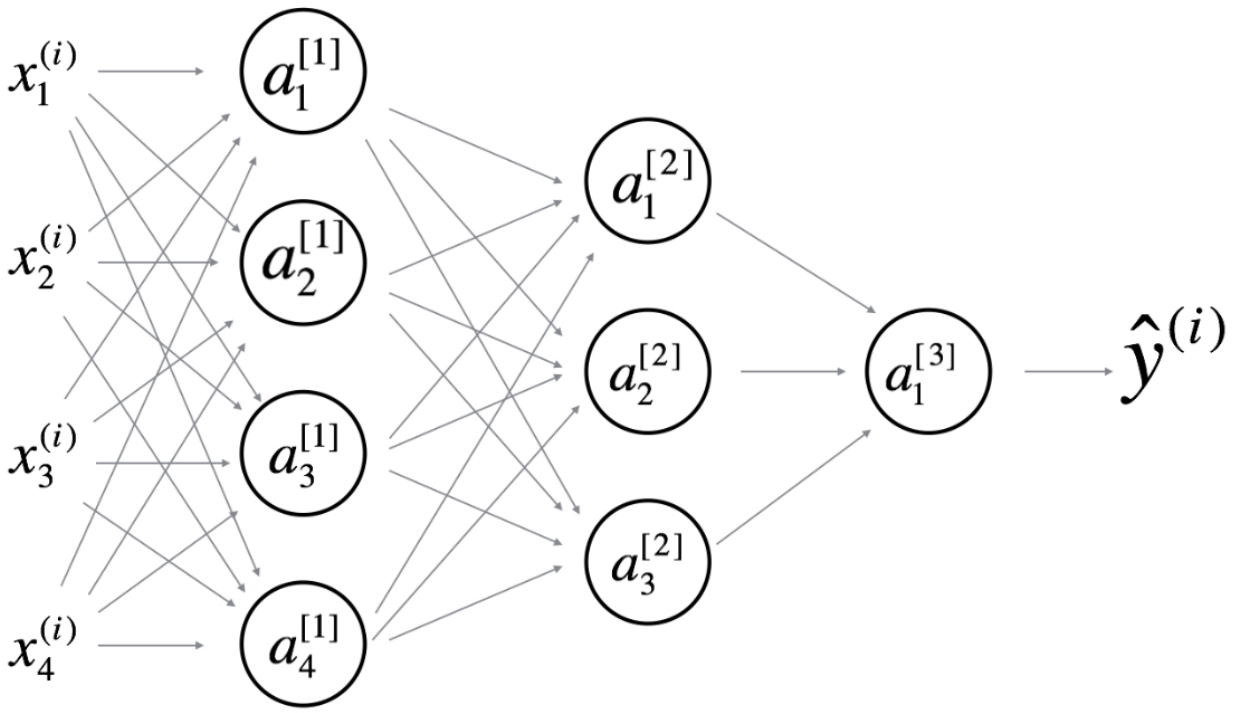
(i) To compute the function using a shallow network circuit, you will need a large network (where we measure size by the number of logic gates in the network), but (ii) To compute it using a deep network circuit, you need only an exponentially smaller network. True/False?

- False
- True

Expand

Correct

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Expand

✓ **Correct**
Great, you got all the right answers.

10. Whereas the previous question used a specific network, in the general case what is the dimension of W^{[l]}, the weight matrix associated with layer l? **1 / 1 point**

- ○ $W^{[l]}$ has shape $(n^{[l-1]}, n^{[l]})$
- ◉ $W^{[l]}$ has shape $(n^{[l]}, n^{[l-1]})$
- ○ $W^{[l]}$ has shape $(n^{[l+1]}, n^{[l]})$
- ○ $W^{[l]}$ has shape $(n^{[l]}, n^{[l+1]})$

Expand

✓ **Correct**
True