Key Concepts on Deep Neural Networks

1. Which of the following is stored in the 'cache' during forward propagation for latter use in backward propagation?

1 / 1 point

- (selected) $Z^{[l]}$
- $W^{[l]}$
- $b^{[l]}$

Expand

Correct
Yes. This value is useful in the calculation of $dW^{[l]}$ in the backward propagation.

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

1 / 1 point

- [ ] $g^{[l]}$ the activation functions.
- [x] $W^{[l]}$ the weight matrices.

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

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

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

- [ ] L the number of layers of the neural network.

Expand

Correct
Great, you got all the right answers.

3. Which of the following statements is true?

1 / 1 point

- (selected) The deeper layers of a neural network are typically computing more complex features of the input than the earlier layers.
- The earlier layers of a neural network are typically computing more complex features of the input than the deeper layers.

Expand

Correct

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
- (selected) False

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.

4. Vectorization allows us to compute $a^{[l]}$ for all the examples on a batch at the same time without using a for loop. True/False?

1 / 1 point

- False
- (selected) True

Expand

Correct

Correct. Vectorization allows us to compute the activation for all the training examples at the same time, avoiding the use of a for loop.

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

- (selected)
```
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
```

-
```
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
```

Expand

Correct

5. Suppose W[i] is the array with the weights of the i-th layer, b[i] is the vector of biases of the i-th layer, and g is the activation function used in all layers. Which of the following calculates the forward propagation for the neural network with L layers.

1 / 1 point

- (selected)
```
for i in range(1, L+1):
Z[i] = W[i]*A[i-1] + b[i]
A[i] = g(Z[i])
```

-
```
for i in range(L):
Z[i+1] = W[i+1]*A[i+1] + b[i+1]
A[i+1] = g(Z[i+1])
```

-
```
for i in range(L):
Z[i] = W[i]*X + b[i]
A[i] = g(Z[i])
```

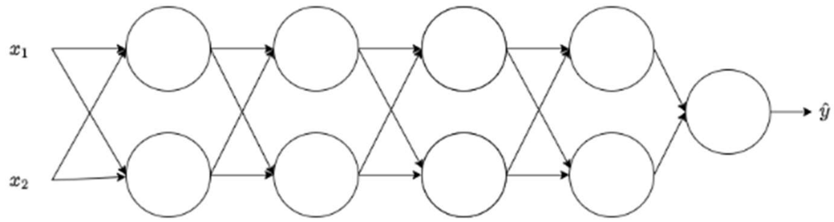
-
```
for i in range(1, L):
Z[i] = W[i]*A[i-1] + b[i]
A[i] = g(Z[i])
```

Expand

Correct
Yes. Remember that the range omits the last number thus the range from 1 to L+1 gives the L necessary values.

6. Consider the following neural network:

1 / 1 point



How many layers does this network have?

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

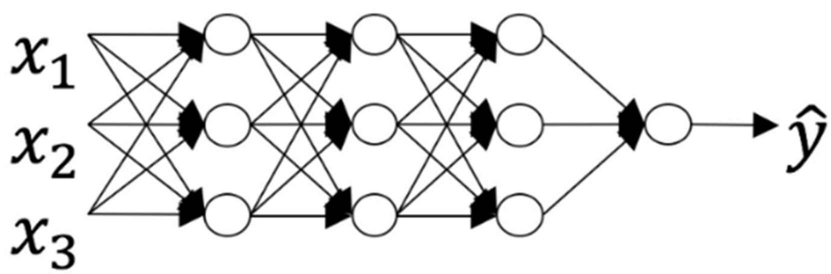
Expand

Correct

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

6. Consider the following neural network.

1 / 1 point



How many layers does this network have?

- ○ The number of layers L is 3. The number of hidden layers is 3.
- ○ The number of layers L is 5. The number of hidden layers is 4.
- ○ 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, to calculate $A^{[l]}$, you use the activation function $g^{[l]}$ with the values of $Z^{[l]}$.

1 / 1 point

True/False: During backward propagation, you calculate $dA^{[l]}$ from $Z^{[l]}$.

- ○ 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 $W^{[l]}$.

7. During forward propagation, in the forward function for a layer l you need to know what is the activation function in a layer (sigmoid, tanh, ReLU, etc.). During backpropagation, the corresponding backward function also needs to know what is the activation function for layer l, since the gradient depends on it. True/False?

1 / 1 point

- False
- True

Expand

Correct
Yes, as you've seen in week 3 each activation has a different derivative. Thus, during backpropagation you need to know which activation was used in the forward propagation to be able to compute the correct derivative.

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?

- True
- False

Expand

Correct

8. A shallow neural network with a single hidden layer and 6 hidden units can compute any function that a neural network with 2 hidden layers and 6 hidden units can compute. True/False?

1 / 1 point

- (•) False
- ( ) True

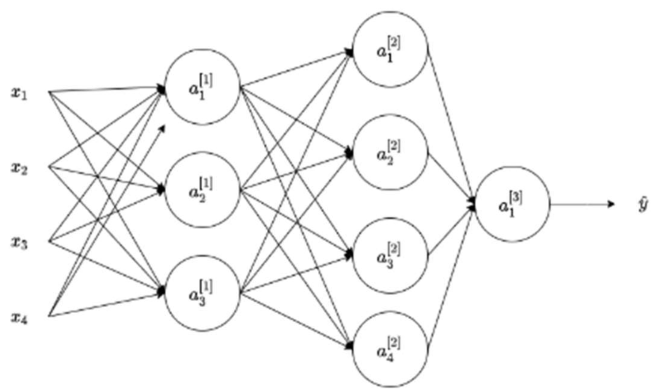
Expand

Correct

Correct. As seen during the lectures there are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.

9. Consider the following 2 hidden layers neural network:

0 / 1 point



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

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

  ✓ **Correct**

  Yes. More generally, the shape of $W^{[l]}$ is $(n^{[l]}, n^{[l-1]})$.
- [x] $b^{[1]}$ will have shape (3, 1)

  ✓ **Correct**

  Yes. More generally, the shape of $b^{[l]}$ is $(n^{[l]}, 1)$.
- [ ] $W^{[2]}$ will have shape (3, 4)
- [ ] $W^{[1]}$ will have shape (3, 4)
- [ ] $b^{[1]}$ will have shape (4, 1)
- [ ] $W^{[2]}$ will have shape (1, 3)
- [ ] $W^{[2]}$ will have shape (3, 1)
- [ ] $W^{[1]}$ will have shape (4, 3)

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

1 / 1 point

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

Expand

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