. ¿Para qué se usa el "caché" en nuestra implementación de propagación hacia adelante y hacia atrás?

1 / 1 punto

● Lo usamos para pasar las variables calculadas durante la propagación hacia adelante al paso de propagación hacia atrás correspondiente. Contiene valores útiles para la propagación hacia atrás para calcular derivadas.

○ Lo usamos para pasar las variables calculadas durante la propagación hacia atrás al paso de propagación hacia adelante correspondiente. Contiene valores útiles para la propagación hacia adelante para calcular activaciones.

○ Se utiliza para almacenar en caché los valores intermedios de la función de costo durante el entrenamiento.

○ It is used to keep track of the hyperparameters that we are searching over, to speed up computation.

✓ **Correcto**
Correct, the "cache" records values from the forward propagation units and sends it to the 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 punto

□ activation values $a^{[l]}$

☑ number of layers L in the neural network

✓ **Correcto**

☑ number of iterations

✓ **Correcto**

☑ learning rate α

✓ **Correcto**

□ bias vectors $b^{[l]}$

□ weight matrices $W^{[l]}$

☑ size of the hidden layers $n^{[l]}$

✓ **Correcto**

3. Which of the following statements is true?

1 / 1 punto

● 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.

✓ **Correcto**

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 punto**

○ True

● False

✓ **Correcto**
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, 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 punto**

○
```
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
```

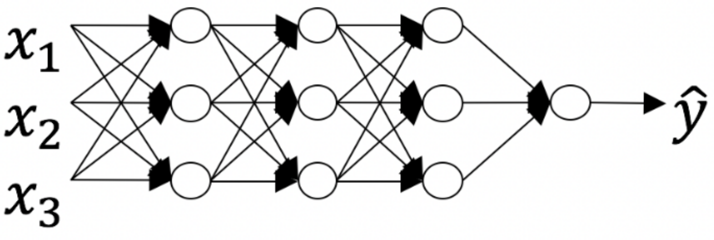
○
```
for i in range(1, len(layer_dims)/2):
    parameter['W' + str(i)] = np.random.randn(layer_dims[i], layer_dims[i-1]) * 0.01
    parameter['b' + str(i)] = np.random.randn(layer_dims[i-1], 1) * 0.01
```

○
```
for i in range(1, len(layer_dims)):
    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)):
    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
```

✓ **Correcto**

6. Consider the following neural network. **1 / 1 punto**



How many layers does this network have?

● The number of layers L is 4. The number of hidden layers is 3.

○ 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 5. The number of hidden layers is 4.

Correcto

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, 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 punto

● True

○ False

Correcto

Yes, as you've seen in the 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 punto

(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?

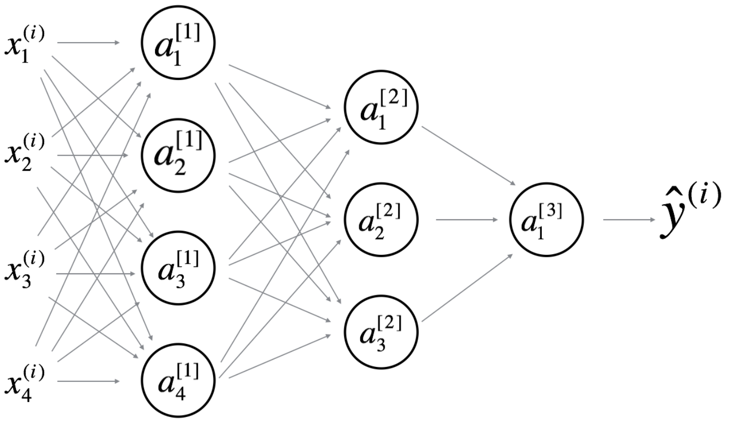
● True

○ False

Correcto

9. Considere la siguiente red neuronal de 2 capas ocultas:

1 / 1 punto



¿Cuáles de las siguientes afirmaciones son verdaderas? (Marque todo lo que corresponda).

☑ $En^{[1]}$tendrá forma (4, 4)

Correcto

Sí. Más generalmente, la forma de$En^{[l]}$es($n^{[l]}, norte^{[l-1]}$).

☑ $b^{[1]}$tendrá forma (4, 1)

✓ **Correcto**
Sí. Más generalmente, la forma de$b^{[l]}$es($n^{[l]}, 1)$.

☐ $En^{[1]}$tendrá forma (3, 4)

☐ $b^{[1]}$tendrá forma (3, 1)

☑ $En^{[2]}$tendrá forma (3, 4)

✓ **Correcto**
Sí. Más generalmente, la forma de$En^{[l]}$es($n^{[l]}, norte^{[l-1]})$.

☐ $b^{[2]}$tendrá forma (1, 1)

☐ $En^{[2]}$tendrá forma (3, 1)

☑ $b^{[2]}$tendrá forma (3, 1)

✓ **Correcto**
Sí. Más generalmente, la forma de$b^{[l]}$es($n^{[l]}, 1)$.

☐ $En^{[3]}$tendrá forma (3, 1)

☑ $b^{[3]}$tendrá forma (1, 1)

✓ **Correcto**
Sí. Más generalmente, la forma de$b^{[l]}$es($n^{[l]}, 1)$.

☑ $En^{[3]}$tendrá forma (1, 3)

✓ **Correcto**
Sí. Más generalmente, la forma de$En^{[l]}$es($n^{[l]}, norte^{[l-1]})$.

☐ $b^{[3]}$tendrá forma (3, 1)

10. Mientras que la pregunta anterior utilizó una red específica, en el caso general, ¿cuál es la dimensión de W^{[l]}, la matriz de peso asociada con la capayo?

1 / 1 punto

○ $En^{[l]}$tiene forma($n^{[l+1]}, norte^{[l]})$

○ $En^{[l]}$tiene forma($n^{[l-1]}, norte^{[l]})$

○ $En^{[l]}$tiene forma($n^{[l]}, norte^{[l+1]})$

● $En^{[l]}$tiene forma($n^{[l]}, norte^{[l-1]})$

✓ **Correcto**
Verdadero