축하합니다! 통과하셨습니다!

받은 학점 100% 최신 제출물 학점 100% 통과 점수: 80% 이상

다음 항목으로 이동

1. What is stored in the 'cache' during forward propagation for latter use in backward propagation? 1 / 1점

- $A^{[l]}$
- $Z^{[l]}$
- $W^{[l]}$
- $b^{[l]}$

더 보기

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

2. Among the following, which ones are "hyperparameters"? (Check all that apply.) 1 / 1점

- [ ] weight matrices $W^{[l]}$
- [x] learning rate α
  - Correct
- [x] number of iterations
  - Correct
- [ ] bias vectors $b^{[l]}$
- [x] size of the hidden layers $n^{[l]}$
  - Correct
- [ ] activation values $a^{[l]}$
- [x] number of layers L in the neural network
  - Correct

더 보기

맞습니다
Great, you got all the right answers.

3. Considering the intermediate results below, which layers of a deep neural network are they likely to belong to? 1 / 1점

- Later layers of the deep neural network.
- Early layers of the deep neural network.
- Middle layers of the deep neural network.
- Input layer of the deep neural network.

더 보기

맞습니다
Correct. The deep layers of a neural network are typically computing more complex features such as the ones shown in the figure.

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 / 1점

- ○ True
- ◉ False

더 보기

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

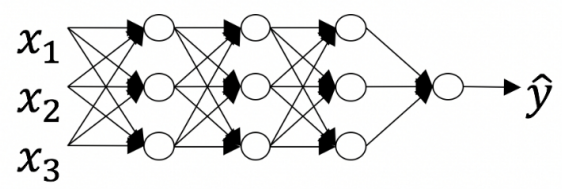
5. Assume we store the values for $n^{[l]}$ in an array called layer_dims, as follows: layer_dims = [n_x, 4,3,2,1]. So layer 1 has four hidden units, layer 2 has 3 hidden units, and so on. Which of the following for-loops will allow you to initialize the parameters for the model? 1 / 1점

- ◉ 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

더 보기

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

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



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 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 5. The number of hidden layers is 4.

더 보기

맞습니다
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. If L is the number of layers of a neural network then $dZ^{[L]} = A^{[L]} - Y$. True/False? 1 / 1점

- ○ False
  No. The gradient of the output layer depends on the difference between the value computed during the forward propagation process and the target values.
- ◉ True
  Yes. The gradient of the output layer depends on the difference between the value computed during the forward propagation process and the target values.

더 보기

맞습니다

8. There are certain functions with the following properties: 1 / 1점

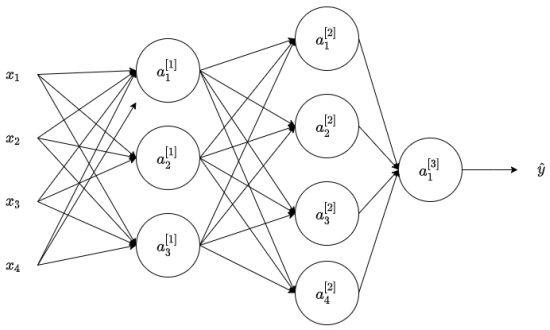
(i) To compute the function using a shallow network circuit, you will need a large network (where we measure size by the number of logic gates in the network), but (ii) To compute it using a deep network circuit, you need only an exponentially smaller network. True/False?

- False
- True

더 보기

맞습니다

9. Consider the following 2 hidden layers neural network: 1 / 1점



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

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

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

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

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

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

더 보기

맞습니다
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 / 1점

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

더 보기

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