✔ 축하합니다! 통과하셨습니다!

받은 성적 100% 통과 점수: 80% 이상

다음 항목으로 이동

Key Concepts on Deep Neural Networks

최근 제출물 성적 100%

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

1 / 1점

- ○ It is used to keep track of the hyperparameters that we are searching over, to speed up computation.
- ◉ We use it to pass variables computed during forward propagation to the corresponding backward propagation step. It contains useful values for backward propagation to compute derivatives.
- ○ 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.
- ○ It is used to cache the intermediate values of the cost function during training.

> 맞습니다
> 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 / 1점

- [x] learning rate α

> 맞습니다

- [x] number of iterations

> 맞습니다

- [ ] weight matrices $W^{[l]}$
- [x] size of the hidden layers $n^{[l]}$

> 맞습니다

- [ ] bias vectors $b^{[l]}$
- [ ] activation values $a^{[l]}$
- [x] number of layers L in the neural network

> 맞습니다

3. Which of the following statements is true?

1 / 1점

- ◉ 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.

> 맞습니다

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 / 1점

- ○ True
- ◉ False

> 맞습니다
> 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 / 1점

- ◉

```
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) * 0.01

for i in range(1, len(layer_dims)/2):

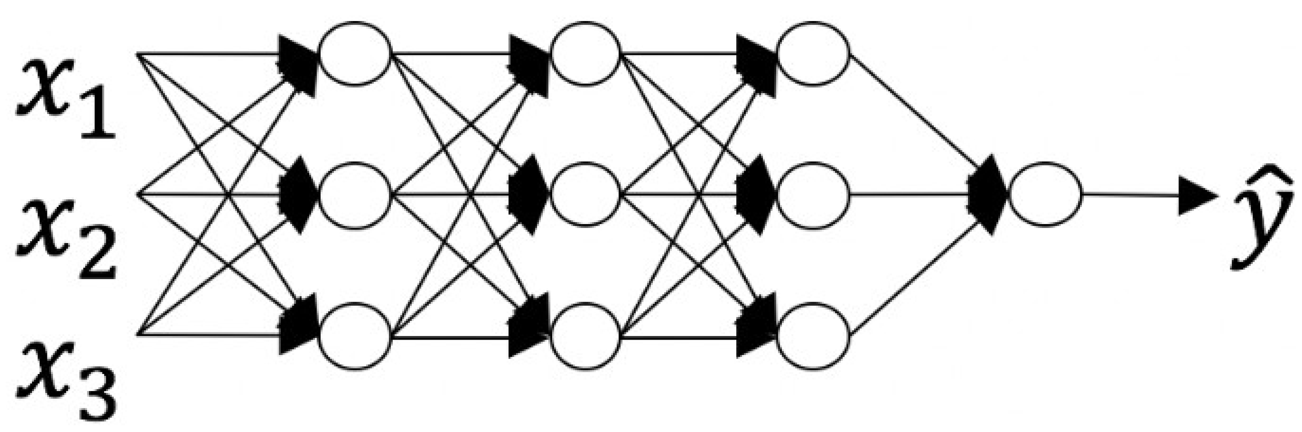
parameter['W' + str(i)] = np.random.randn(layer_dims[i], layer_dims[i-1]) * 0.01

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

맞습니다

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.
- (selected) 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.

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. 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 / 1점

- (selected) True
- False

맞습니다
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 / 1점

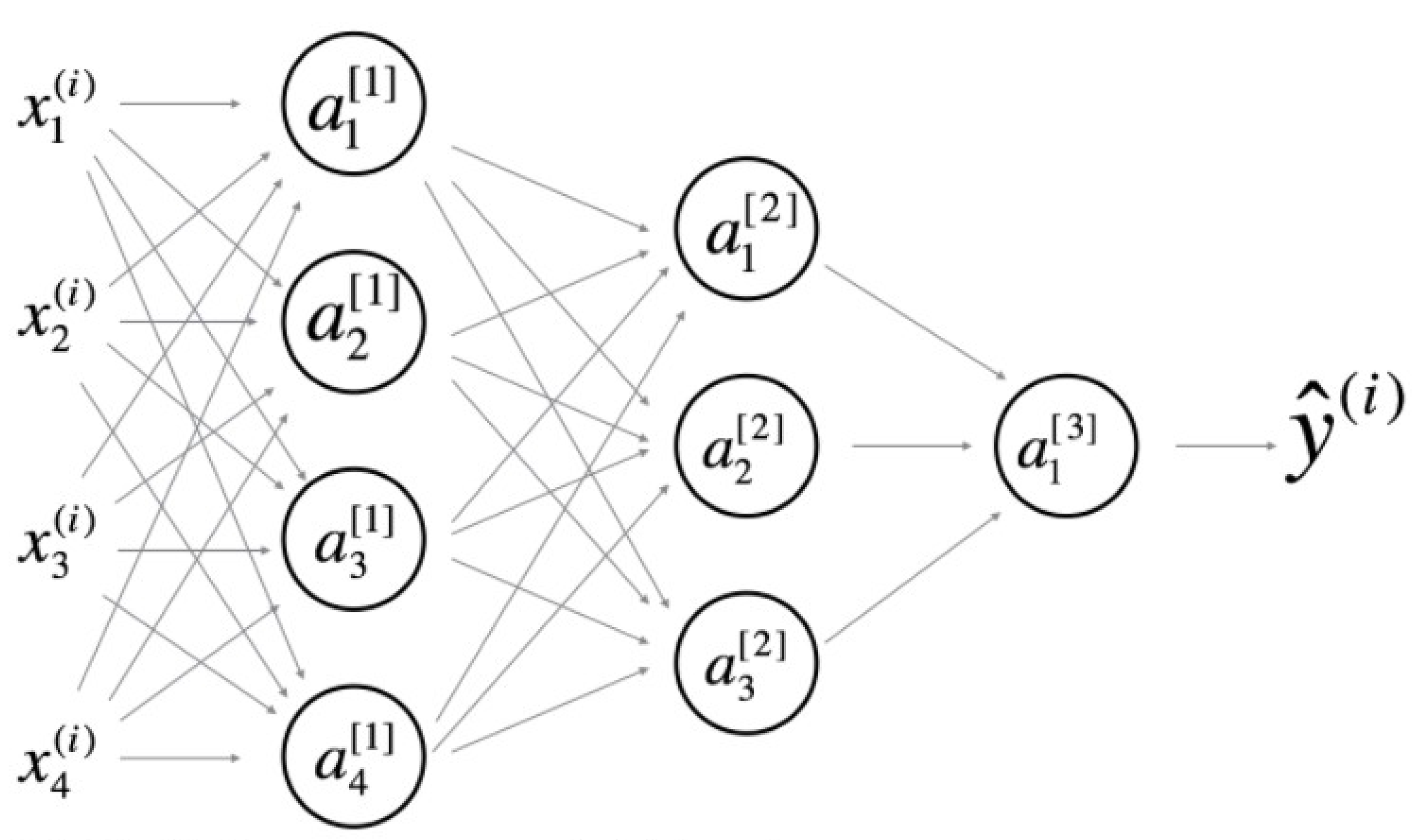
(i) To compute the function using a shallow network circuit, you will need a large network (where we measure size by the number of logic gates in the network), but (ii) To compute it using a deep network circuit, you need only an exponentially smaller network. True/False?

- (selected) True
- False

맞습니다

9. Consider the following 2 hidden layer neural network:

1 / 1점



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

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

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

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

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

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

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

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

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

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

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

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

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

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 / 1점

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

맞습니다
True