← Key concepts on Deep Neural Networks Graded Quiz ∙ 30 min Due Feb 1, 3:59 PM CST

Key concepts on Deep Neural Networks

Submit your assignment DUE DATE Feb 1, 3:59 PM CST ATTEMPTS 3 every 8 hours

Video: Deep L-layer neural network 5 min

Reading: Clarification about Getting your matrix dimensions right video 1 min

Video: Getting your matrix dimensions right 11 min

Reading: Clarification about What does this have to do with the brain video 1 min

Video: What does this have to do with the brain?
3 min

Practice Questions Quiz: Key concepts on Deep Neural Networks 10 questions

Video: Why deep representations?

Receive grade Grade
100%

View Feedback
We keep your highest score TO PASS 80% or higher

Video: Building blocks of deep neural networks
8 min Reading: Clarification about Upcoming Forward and Backward Propagation Video 1 min Video: Forward and Backward Propagation 10 min

6 P P

✓ Congratulations! You passed! Keep Learning GRADE 100% TO PASS 80% or higher **Key concepts on Deep Neural Networks** 1. What is the "cache" used for in our implementation of forward propagation and backward propagation? O 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 backward propagation to the corresponding forward propagation step. It contains useful values for forward propagation to compute activations. 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.) lacksquare size of the hidden layers $n^{[l]}$ ✓ Correct $oxed{\ }$ bias vectors $b^{[l]}$ ✓ Correct number of iterations lacksquare learning rate lphaThe 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. 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? False ✓ Correct 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: $(a^{[i]}=g^{[i]}(z^{[i]}),z^{[i]}=W^{[i]}a^{[i-1]}+b^{[i]},...)$. 5. Assume we store the values for $n_i^{[l]}$ 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? 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['M' + 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 1 for i in range(1, len(layer_dims)):
2 parameter['W' + str(i)] = np.random.randn(layer_dims[i-1], layer_dims[i]) * 0.01
3 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 6. Consider the following neural network. $\ensuremath{\bigodot}$ The number of layers L is 4. The number of hidden layers is 3. igcap The number of layers L is 3. The number of hidden layers is 3. igcup The number of layers L is 4. The number of hidden layers is 4. igcup The number of layers L is 5. The number of hidden layers is 4. 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. 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? True ○ False 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: (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 ✓ Correct 9. Consider the following 2 hidden layer neural network: Which of the following statements are True? (Check all that apply). $lacksquare W^{[1]}$ will have shape (4, 4) Yes. More generally, the shape of $W^{[l]}$ is $(n^{[l]}, n^{[l-1]})$. $b^{[1]}$ will have shape (4, 1) \checkmark Correct Yes. More generally, the shape of $b^{[l]}$ is $(n^{[l]},1).$ $b^{[1]}$ will have shape (3, 1) $lacksquare W^{[2]}$ will have shape (3, 4) \checkmark Correct Yes. More generally, the shape of $W^{[l]}$ is $(n^{[l]}, n^{[l-1]}).$ $\ \ \ \ \ b^{[2]}$ will have shape (1, 1) $lacksquare b^{[2]}$ will have shape (3, 1) ✓ Correct Yes. More generally, the shape of $b^{[l]}$ is $(n^{[l]},1)$. $lacksquare b^{[3]}$ will have shape (1, 1) \checkmark Correct Yes. More generally, the shape of $b^{[l]}$ is $(n^{[l]},1).$ $lacksquare W^{[3]}$ will have shape (1, 3)

 \checkmark Correct 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^{[]]}, the weight matrix associated with layer *l*?

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

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