Week 4

Deep L-Layer neural network

- $L = 4$ (# of layers)
- $n^{[l]}$ = #units in layers l
- $n^{[0]} = 3$ (input layer) $, n^{[1]} = 5, n^{[2]} = 5, n^{[3]} = 3, n^{[4]} = 1$ (output layer)
- $a^{[l]}$ (activation in layer l)
- $a^{[l]} = g^{[l]}(z^{[l]}), w^{[l]} =$ weights for $z^{[l]}, b^{[l]} =$ bias for $z^{[l]}$
- $a^{[4]} = \hat{y}$

Forward Propagation in a Deep Neural Network

- $Z^{[l]} = W^{[l]} A^{[l-1]} + B^{[l]}$
- $A^{[l]} = g^{[l]}(A^{[l]})$

Getting your matrix dimension right

- $Z^{[l]}.\text{shape} = (n^{[l]}, m)$
- $W^{[l]}.\text{shape} = (n^{[l]}, n^{[l-1]})$
- $A^{[l]}.\text{shape} = (n^{[l-1]}, m)$
- $dW^{[l]}.\text{shape} = (n^{[l]}, n^{[l-1]})$
- $db^{[l]}.\text{shape} = (n^{[l]}, m)$

Why deep representations

- Compositional representation: Shallow networks are able to detect simple features, deep layers are able to detect complex functions and are able to model much more complex data from the simple features
- Circuit theory: There are functions you can compute with a “small” L-layer deep neural network that shallower networks require exponentially more hidden units to compute. e.g.: XOR detection: 2 layer 3-2-1 neurons vs 1 layer with 2^n neurons to map all the combinations of the inputs

Parameter vs Hyperparameter

- Parameters:
  - Weights
  - Biases
- Hyperparameters:
  - Learning rate α or $f(t) = \theta$
  - # iterations
  - # hidden units
  - choice of activation function
  - Momentum
  - Mini-batch size
  - Regularization

Quiz

1. What is the “cache” used for in our implementation of forward propagation and backward propagation?
   - [ ] 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.
   - [x] 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.
   - [ ] It is used to keep track of the hyperparameters that we are searching over, to speed up computation.
   - [ ] It is used to cache the intermediate values of the cost function during training.
2. Among the following, which ones are “hyperparameters”? (Check all that apply.)
   - [ ] activation values $a^{[1]}$
   - [x] number of iterations
   - [ ] weight matrices $W^{[1]}$
   - [x] number of layers L in the neural network
   - [x] learning rate α
   - [x] size of the hidden layers $n^{[1]}$
   - [ ] bias vectors $b^{[1]}$
3. Which of the following statements is true?
   - [x] 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
4. Vectorization allows you to compute forward propagation in an LL-layer neural network without an explicit for-loop (or any other explicit iterative loop) over the layers l=1, 2, …,L. True/False?
   - [ ] True
   - [x] False
5. Assume we store the values for n^{[l]} in an array called layers, 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?
   - [ ] `python for(i in range(1, len(layer_dims)/2)): parameter[‘W’ + str(i)] = np.random.randn(layers[i], layers[i-1])) * 0.01 parameter[‘b’ + str(i)] = np.random.randn(layers[i], 1) * 0.01`
   - [ ] `python for(i in range(1, len(layer_dims)/2)): parameter[‘W’ + str(i)] = np.random.randn(layers[i], layers[i-1])) * 0.01 parameter[‘b’ + str(i)] = np.random.randn(layers[i-1], 1) * 0.01`
   - [ ] `python for(i in range(1, len(layer_dims))): parameter[‘W’ + str(i)] = np.random.randn(layers[i-1], layers[i])) * 0.01 parameter[‘b’ + str(i)] = np.random.randn(layers[i], 1) * 0.01`
   - [x] `python for(i in range(1, len(layer_dims))): parameter[‘W’ + str(i)] = np.random.randn(layers[i], layers[i-1])) * 0.01 parameter[‘b’ + str(i)] = np.random.randn(layers[i], 1) * 0.01`
6. Consider the following neural network. How many layers does this network have?
   - [x] 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.

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?
   - ☑ True
   - ☐ False
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
9. Consider the following 2 hidden layer neural network: Which of the following statements are True? (Check all that apply).
   - ☑ $W^{[1]}$ will have shape (4, 4)
   - ☑ $b^{[1]}$ will have shape (4, 1)
   - ☐ $W^{[1]}$ will have shape (3, 4)
   - ☐ $b^{[1]}$ will have shape (3, 1)
   - ☑ $W^{[2]}$ will have shape (3, 4)
   - ☐ $b^{[2]}$ will have shape (1, 1)
   - ☐ $W^{[2]}$ will have shape (3, 1)
   - ☑ $b^{[2]}$ will have shape (3, 1)
   - ☐ $W^{[3]}$ will have shape (3, 1)
   - ☑ $b^{[3]}$ will have shape (1, 1)
   - ☑ $W^{[3]}$ will have shape (1, 3)
   - ☐ $b^{[3]}$ will have shape (3, 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 ll?
   - ☐ $W^{[l]}$ has shape $(n^{[l-1]}, n^{[l]})$
   - ☑ $W^{[l]}$ has shape $(n^{[l]}, n^{[l-1]})$
   - ☐ $W^{[l]}$ has shape $(n^{[l]}, n^{[l+1]})$
   - ☐ $W^{[l]}$ has shape $(n^{[l+1]}, n^{[l]})$