Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

Next item →

1. Which of the following are true? (Check all that apply.) 1 / 1 point

- [ ] $a_4^{[2]}$ is the activation output of the 2^{nd} layer for the 4^{th} training example
- [ ] X is a matrix in which each row is one training example.
- [x] $a^{[2](12)}$ denotes the activation vector of the 2^{nd} layer for the 12^{th} training example.

Correct

- [x] X is a matrix in which each column is one training example.

Correct

- [x] $a_4^{[2]}$ is the activation output by the 4^{th} neuron of the 2^{nd} layer

Correct

- [ ] $a^{[2](12)}$ denotes activation vector of the 12^{th} layer on the 2^{nd} training example.
- [x] $a^{[2]}$ denotes the activation vector of the 2^{nd} layer.

Correct

2. The sigmoid function is only mentioned as an activation function for historical reasons. The tanh is always preferred without exceptions in all the layers of a Neural Network. True/False? 1 / 1 point

- (•) False
- ( ) True

Correct
Yes. Although the tanh almost always works better than the sigmoid function when used in hidden layers, thus is always proffered as activation function, the exception is for the output layer in classification problems.

3. Which of these is a correct vectorized implementation of forward propagation for layer l, where $1 \leq l \leq L$? 1 / 1 point

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

Correct

4. The use of the ReLU activation function is becoming more rare because the ReLU function has no derivative for $c = 0$. True/False? 1 / 1 point

- (•) False
- ( ) True

Correct
Yes. Although the ReLU function has no derivative at $c = 0$ this rarely causes any problems in practice. Moreover it has become the default activation function in many cases, as explained in the lectures.

5. Consider the following code: 1 / 1 point

```
#+begin_src python
x = np.random.rand(3, 2)
y = np.sum(x, axis=0, keepdims=True)
#+end_src
```

What will be y.shape?

- (2,)
- (1, 2) ✓
- (3, 1)
- (3,)

Correct
Yes. By choosing the axis=0 the sum is computed over each column of the array, thus the resulting array is a row vector with 2 entries. Since the option keepdims=True is used the first dimension is kept, thus (1, 2).

6. Suppose you have built a neural network with one hidden layer and tanh as activation function for the hidden layers. Which of the following is a best option to initialize the weights? 1 / 1 point

- Initialize all weights to a single number chosen randomly.
- Initialize all weights to 0.
- Initialize the weights to small random numbers. ✓
- Initialize the weights to large random numbers.

Correct
The use of random numbers helps to "break the symmetry" between all the neurons allowing them to compute different functions. When using small random numbers the values $z^{[k]}$ will be close to zero thus the activation values will have a larger gradient speeding up the training process.

7. Using linear activation functions in the hidden layers of a multilayer neural network is equivalent to using a single layer. True/False? 1 / 1 point

- True ✓
- False

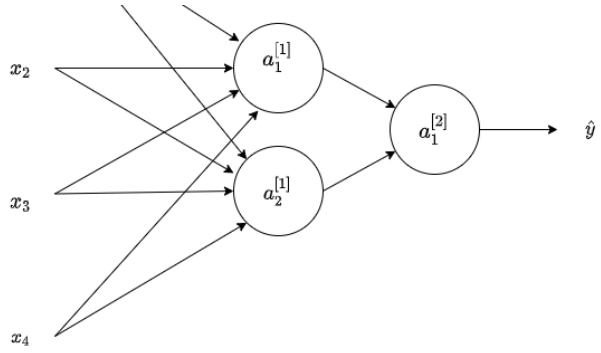
Correct
Yes. When the identity or linear activation function $g(c) = c$ is used the output of composition of layers is equivalent to the computations made by a single layer.

8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relatively large values, using np.random.randn(..,..)*1000. What will happen? 1 / 1 point

- So long as you initialize the weights randomly gradient descent is not affected by whether the weights are large or small.
- This will cause the inputs of the tanh to also be very large, thus causing gradients to be close to zero. The optimization algorithm will thus become slow. ✓
- This will cause the inputs of the tanh to also be very large, thus causing gradients to also become large. You therefore have to set α to a very small value to prevent divergence; this will slow down learning.
- This will cause the inputs of the tanh to also be very large, causing the units to be “highly activated” and thus speed up learning compared to if the weights had to start from small values.

Correct
Yes. tanh becomes flat for large values; this leads its gradient to be close to zero. This slows down the optimization algorithm.

9. Consider the following 1 hidden layer neural network: 1 / 1 point



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

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

Correct
Yes. The number of rows in $W^{[k]}$ is the number of neurons in the k-th layer and the number of columns is the number of inputs of the layer.

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

Correct
Yes. The number of rows in $W^{[k]}$ is the number of neurons in the k-th layer and the number of columns is the number of inputs of the layer.

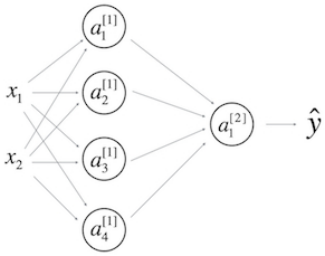
- [x] $b^{[1]}$ will have shape (2, 1).

Correct
Yes. $b^{[k]}$ is a column vector and has the same number of rows as neurons in the k-th layer.

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

10. What are the dimensions of $Z^{[1]}$ and $A^{[1]}$?

1 / 1 point



- (•) $Z^{[1]}$ and $A^{[1]}$ are (4,m)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (4,2)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (4,1)
- ( ) $Z^{[1]}$ and $A^{[1]}$ are (1,4)

Correct