Your grade: 88.33%

Your latest: **88.33%** • Your highest: **88.33%** • 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.) 0.8333333333333334 / 1 point

- [ ] $w_3^{[4]}$ is the row vector of parameters of the fourth layer and third neuron.
- [x] $w_3^{[4]}$ is the column vector of parameters of the fourth layer and third neuron.

Correct
Yes. The vector $w_j^{[i]}$ is the column vector of parameters of the i-th layer and j-th neuron of that layer.

- [ ] $a_3^{[2]}$ denotes the activation vector of the second layer for the third example.
- [x] $a^{[2]}$ denotes the activation vector of the second layer.

Correct
Yes. In our convention $a^{[j]}$ denotes the activation function of the j-th layer.

- [x] $w_3^{[4]}$ is the column vector of parameters of the third layer and fourth neuron.

This should not be selected
No. The vector $w_j^{[i]}$ is the column vector of parameters of the jth neuron in the i-th layer.

- [ ] $a^{[3](2)}$ denotes the activation vector of the second layer for the third example.

2. The tanh activation is not always better than sigmoid activation function for hidden units because the mean of its output is closer to zero, and so it centers the data, making learning complex for the next layer. True/False? 1 / 1 point

- (•) False
- ( ) True

Correct
Yes. As seen in lecture the output of the tanh is between -1 and 1, it thus centers the data which makes the learning simpler for the next layer.

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

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

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 large random numbers.
- ◉ Initialize the weights to small 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. A single output and single layer neural network that uses the sigmoid function as activation is equivalent to the logistic regression. True/False **1 / 1 point**

- ○ False
- ◉ True

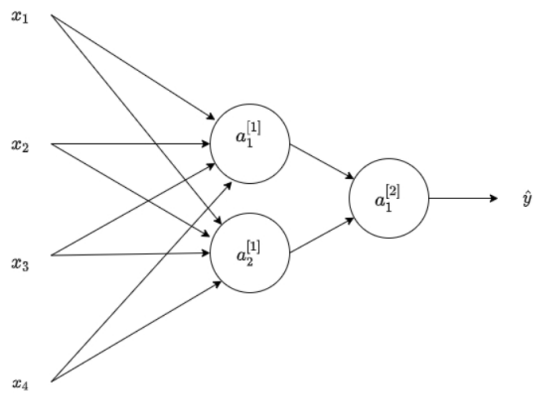
Correct
Yes. The logistic regression model can be expressed by $\hat{y} = \sigma\left(W\,x + b\right)$. This is the same as $a^{[1]} = \sigma(W^{[1]}\,X + b)$.

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**

- ○ 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, 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, causing the units to be “highly activated” and thus speed up learning compared to if the weights had to start from small values.
- ○ So long as you initialize the weights randomly gradient descent is not affected by whether the weights are large or small.

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).

- [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.

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

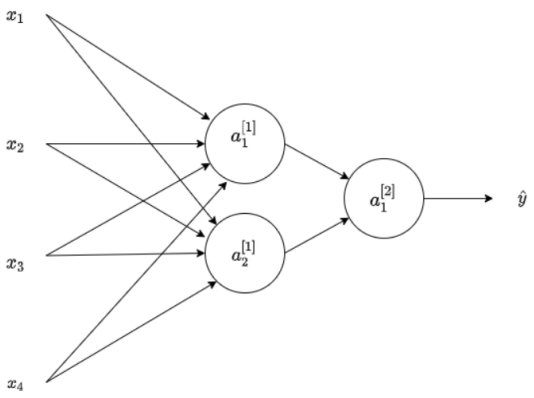
> **Correct**
> Yes. The number of rows in $W^{[k]}$ is the number of neurons in the k-th layer and the number of columns is the number of inputs of the layer.

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

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

10. Consider the following 1 hidden layer neural network: **0 / 1 point**



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

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

> **Incorrect**
> No. The $Z^{[1]}$ and $A^{[1]}$ are calculated over a batch of training examples. The number of columns in $Z^{[1]}$ and $A^{[1]}$ is equal to the number of examples in the batch, m. And the number of rows in $Z^{[1]}$ and $A^{[1]}$ is equal to the number of neurons in the first layer.