Congratulations! You passed!

Grade received 80% **Latest Submission Grade** 80% **To pass** 80% or higher

Go to next item

1. Which of the following are true? (Check all that apply.) **0 / 1 point**

- [ ] $W^{[1]}$ is a matrix with rows equal to the parameter vectors of the first layer.
- [ ] W_1 is a matrix with rows equal to the parameter vectors of the first layer.
- [x] $W^{[1]}$ is a matrix with rows equal to the transpose of the parameter vectors of the first layer.

  ✓ **Correct**
  Yes. We construct $W^{[1]}$ stacking the parameter vectors $w_j^{[1]}$ of all the neurons of the first layer.

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

  ! **This should not be selected**
  No. The vectors $w_k^{[j]}$ are column vectors.

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

Expand

⊗ **Incorrect**
You didn't select all the correct answers

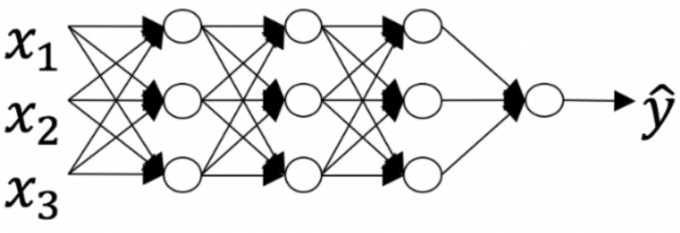
2. The tanh activation is not always better than sigmoid activation function for hidden units because the mean of its output is closer to zero, and so it centers the data, making learning complex for the next layer. True/False? **1 / 1 point**

- ○ True
- ◉ False

Expand

✓ **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 the following represents the activation output of the second neuron of the third layer applied to the fourth example? **1 / 1 point**



- $a_2^{[3](4)}$
- $a_2^{[4](3)}$
- $a_4^{[3](2)}$
- $a_3^{[4]2}$

Expand

Correct
Yes. The superscript in brackets indicates the layer number, the superscript in parenthesis represents the number of examples, and the subscript the number of the neuron.

4. You are building a binary classifier for recognizing cucumbers (y=1) vs. watermelons (y=0). Which one of these activation functions would you recommend using for the output layer? **1 / 1 point**

- Leaky ReLU
- sigmoid
- tanh
- ReLU

Expand

Correct
Yes. Sigmoid outputs a value between 0 and 1 which makes it a very good choice for binary classification. You can classify as 0 if the output is less than 0.5 and classify as 1 if the output is more than 0.5. It can be done with tanh as well but it is less convenient as the output is between -1 and 1.

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?

- (1, 2)
- (3,)
- (3, 1)
- (2,)

Expand

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

Expand

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

- (selected) True
- False

Expand

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

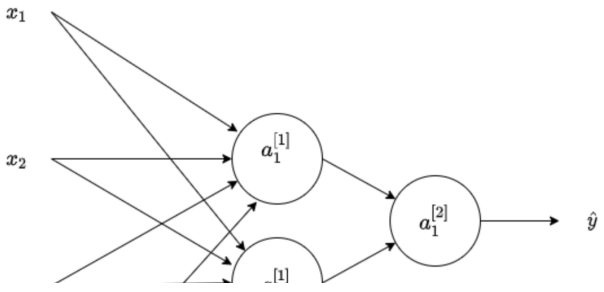
- This will cause the inputs of the tanh to also be very large, thus causing gradients to also become large. You therefore have to set α to a very small value to prevent divergence; this will slow down learning.
- (selected) This will cause the inputs of the tanh to also be very large, thus causing gradients to be close to zero. The optimization algorithm will thus become slow.
- 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, causing the units to be “highly activated” and thus speed up learning compared to if the weights had to start from small values.

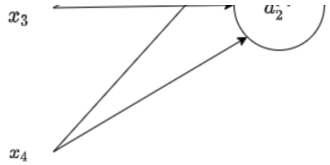
Expand

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.

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

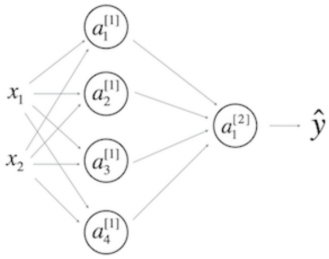
- [ ] $W^{[2]}$ will have shape (2, 1)
- [ ] $W^{[1]}$ will have shape (4, 2).

Expand

Correct
Great, you got all the right answers.

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

0 / 1 point



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

Expand

Incorrect
Remember that $Z^{[1]}$ and $A^{[1]}$ are quantities computed over a batch of training examples, not only 1.