✓ Congratulations! You passed!



## **Shallow Neural Networks**

LATEST SUBMISSION GRADE	

	LATEST SUBMISSION GRADE 100%			
1.	Which of the following are true? (Check all that apply.) $a_{1}^{(2)} \text{ is the activation output of the } 2^{nd} \text{ layer for the } 4^{th} \text{ training example}$	1/1 point		
	$\begin{tabular}{ll} $a^{[1]}$ denotes the activation vector of the 2^{nd} layer.$			
	✓ Correct			
	$\boxed{\mbox{\ensuremath{\mathbb{Z}}}}  a_4^{[2]}$ is the activation output by the $4^{th}$ neuron of the $2^{nd}$ layer			
	✓ Correct			
	$\begin{tabular}{ll} \hline & X \mbox{ is a matrix in which each row is one training example.} \end{tabular}$			
	$\hfill \hfill $			
	$\ensuremath{\mathbb{Z}} \ a^{[j](1)}$ denotes the activation vector of the $2^{nf}$ layer for the $12^{th}$ training example.			
	✓ Correct			
	$\ensuremath{ \swarrow } \ensuremath{ X}$ is a matrix in which each column is one training example.			



(iii) True

○ False

✓ Correct

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  $1 \leq l \leq L?$ 



 $\begin{array}{cccc} & & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & \\ & & & \\ & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ &$ (a) •  $Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$ •  $A^{[l]} = g^{[l]}(Z^{[l]})$  $O \cdot Z^{[l]} = W^{[l]}A^{[l]} + b^{[l]}$   $\cdot A^{[l+1]} = g^{[l]}(Z^{[l]})$ 

✓ Correct



1/1 point

 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? ○ ReLU



O Leaky ReLU ○ tanh



✓ 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 turn as well but it is less convenient as the output is between 1 and 1.



What will be B.shape? (If you're not sure, feel free to run this in python to find out).

(,3) (1, 3) (4, 1)



Suppose you have built a neural network. You decide to initialize the weights and biases to be zero.
Which of the following statements is true?

 Each neuron in the first hidden layer will perform the same computation. So even after multiple iterations of gradient descent each neuron in the layer will be computing the same thing as other neurons. Each neuron in the first hidden layer will perform the same computation in the first iteration. But after one iteration of gradient descent they will learn to compute different things because we have "broken symmetry".

 Each neuron in the first hidden layer will compute the same thing, but neurons in different layers will compute different things, thus we have accomplished "symmetry breaking" as described in lecture. The first hidden layer's neurons will perform different computations from each other even in the first iteration; their parameters will thus keep evolving in their own way.

✓ Correct





False

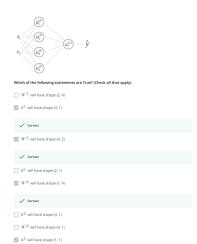


 You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large values, using np.random.randn(....)\*1000. What will happen? This will cause the inputs of the tanh to also be very large, thus causing gradients to be close to zero.
The optimization algorithm will thus become slow. It doesn't matter. So long as you initialize the weights randomly gradient descent is not affected by whether the weights are large or small.

 $\bigcirc$  This will cause the inputs of the tanh to also be very large, thus causing gradients to also be large. You therefore have to set  $\alpha$  to be very small to prevent divergence; this will slow down learning.

Correct
Yes, tanh becomes flat for large values, this leads its gradient to be close to zero. This slows down the optim 9. Consider the following 1 hidden layer neural network:





## 10. In the same network as the previous question, what are the dimensions of $Z^{[1]}$ and $A^{[1]}$ ?

✓ Correct

1/1 point

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

