Shallow Neural Networks    Committee   Com
Total Control
The state of the control of the co
And in a content of the content
□ 18 The content of
In the content of t
Comment of the comm
□ The control of control and the control of the co
The content of the
public content of content and content and the content and th
A company of the co
The continues of the
2. The enter control and an object of the control
A burnary beach of an extension of grant and grant of grant of grant and grant of grant and grant of grant and grant of gr
The control of the co
The control of the co
Compared to the compared to
A 100 App in the act a summa account of a private part of the act
## Section of the Control of the Co
2
• A ** ** ** ** ** ** ** ** ** ** ** ** *
• • • • • • • • • • • • • • • • • • •
• (a) 1 - (a
■ State 1801 *** (10 m s)  ■ A visual to building a body districtive or organing managinary sign and managing sign which are of stress minimum of a visual stress of the state of the sta
4. You will have desired as the control of the cont
On the second production of the second control of the second con
Ancester would get commend unity for the output (specific)    Promp   Fest
O MAZ  O May Service  O start  Victor
<ul> <li>⊕ regrentil</li> <li>☑ Life</li> <li>✓ Careau</li> <li>International Control Section of the control of the control</li></ul>
To provide designate a value is borrown to and indication as a very good of over far bursty consideration. The contraction of t
1. Consider the following company in the first first of the company in the first first first of the company in the first
Consider the following code  1. In a represent restrict, rel  1. In a represent rel  1. In a representation of the first house in restrict rel in a representation of a restrict relation relation relation relation relationship of the restrict relationship of grade in a computing the same relation of the relationship of grade in the computing the same relationship or other relationship  2. In the restrict relationship will be great the computing the same relationship or other relationship  3. In the relationship of the relationship will be great the computing the same relationship or other relationship  3. In the relationship of the relationship will be great the computing the same relationship or other relationship  3. In the relationship of the relationship will be great the restrict restrict relationship or the relationship  3. In the relationship of the relationship or the relationship or the relationship or the relationship  4. Constant  4. The report regressionship or the same is considered by relationship or the restrict relationship  4. The relationship of the relationship or the
What will be a shaped of you're not store. Fed free to run two in python to find out.  (ii) (iii) (iii) (iv) (iv) (iv) (iv) (
What and lose Euthoped (if you've not sure, feel free to run this in potion to find oad)  ( 2)  ( 3)  ( 4)  ( 1)  ( 1)  ( 1)  ( 1)  ( 1)  ( 1)  ( 1)  ( 1)  ( 1)  ( 2)  ( 3)  ( 4)  ( 1)  ( 4)  ( 5)  ( 6)  ( 6)  ( 7)  ( 1)  ( 7)  ( 1)  ( 8)  ( 9)  ( 9)  ( 1)  ( 9)  ( 1)  ( 9)  ( 1)
<ul> <li>○ 1.33</li> <li>○ 14.3</li> <li>○ 14.4</li> <li>○ 14.5</li> <li>○ 14.5</li></ul>
<ul> <li>(4.1)</li> <li>(1.2)</li> <li>(4.1)</li> <li>✓ Carect</li> <li>Yes we use (deepdings = Trust) to make sure that Authors (4.1) it makes our code more repressure.</li> <li>4. Suppose you have built a neveral network, You decide to initiative the weights and biases to be zero. Which of the following:</li></ul>
(1.3)  (4.1)  Cerest  Yes, we use (Recodings - Trust to make our eithat Authorae is (4.1) and rist (4.) it makes our code more rigorous.  6. Supposes you have built a neutral network. You decide to initiative the verigints and bisset to be zero. Which of the following tradements a trust place in the first release of incoming the same fillings and the remarkable incoming the same and the same fillings and the remarkable incoming the same and the same fillings and the remarkable incoming the same with the same fillings and the remarkable incoming the same with the same fillings and the remarkable incoming the same with the same fillings and the remarkable incoming the same and the same fillings and the remarkable incoming the same and the same fillings and the remarkable incoming the same and the same fillings and the remarkable incoming the same and the same fillings and t
Variet  Var. wo use (Response = Trus) to make sure that Authope is (4.1) and not (4.). It makes our code more reporture.  8. Suppose you have built a neural network. You decide to installate the weights and biases to be zero. Which of the following statements is true!  9. Each neuron in the first hidden layer will perform the same computation. So even after multiple statements of graden descent the value in the first hidden layer will be computing the same thing as other resource.  O Each neuron in the first hidden layer will be computing the same thing as other resource.  O Each neuron in the first hidden layer will be computing the same thing as other resource.  O Each neuron in the first hidden layer will be computing the same thing to the same that the state of or gradent descent they will hent to compute different things because or "Ocean" the property.  O Each neuron in the first hidden layer will be reform the same computations of the same tayers will compute officially the same have accomplished symmetry health gains as described in feature.  O The first hidden layer in process will perform the first the computations from each other even in the first iteration their parameters will find to learn a useful decision beundary because it will fall to "break symmetry". True-Palser  Values  • False  Values  **Curret*  Values to the first learned will be first layer in the layer. If you initialize the weights to zeros, the first example x feel in the layers regression diseast have a hidden layer. If you initialize the weights to zeros, the first example x feel in the layers regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x feel in the layers regression will be a to the definition of the computation of the layers to regression depend on the liquid x (because there is not a constant vector, our palse and the definition of the sound on the layer of all the hidden layer. If you initialize the weights to relative large.  **In the first learned to the sound of the pal
Ves. we use (seepdims = "True) to make suite that Authore is (4.1) and not (4.3) it makes our code more rigorous.  6. Supposed you have built a neural network. You decide to initialize the weights and biases to be zero. Which of the following to the substantial of the following the same computation. So even after multiple iterations of gradient descent form while larn to more different things because the true transmitter.  © Each neuron in the first hidden layer will perform the same computation in the first iteration. But after one iteration of gradient descent from little and one of gradient descent the transmitter.  © Tach neuron in the first hidden layer will compute the same thing, but neuron in different buyer will compute different things; because the transmitter.  © The first hidden layer's neurons will perform different computations from each other even in the first, iterations their parameters will than the proving in their own way.  © Correct  7. Logistic regression will fail to learn a useful decision boundary because it will fail to "treas symmetry", "Troef-staze" the logistic regression will fail to learn a useful decision boundary because it will fail to "treas symmetry", "Troef-staze" to lite the layer regression desent have a hidden layer. If you initialize the weights to zeros, the first example x fed in the layes regression desent have a hidden layer. If you initialize the weights to zeros, the first example x fed in the layest regression desent have a hidden layer. If you initialize the weights to zeros, the first example x fed in the layest regression desent have a hidden layer. If you initialize the weights to zeros, the first example x fed in the layer regression will not usual zero but the deniatives of the Logistic Regression depend on the input x (breasses there so hidden you have built a network using the ta
© Each neuron in the first hidden layer will perform the same computation. So even after multiple fereations 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 instantion. But after one iteration of gradient descent they will learn to compute different high because we have "Orden symmetry."  ○ Each neuron in the first hidden layer will compute the same thing; but neurons in offerent layers will compute different things, thus we have accomplished "symmetry breaking" as described in lecture.  ○ The first hidden layers neurons will perform different computations from each other even in the first iteration; their parameters will thus keep evolving in their own way.  ✓ Cerrect  7. Logistic regression's weights w should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to "break symmetry", Truerificials?  ○ True  ○ True  ○ True  ○ False  ✓ Correct  Normal Supposition departs a present point in the environment of the Logistic Regression depart on the layer and the layer and the layer and the layer and the layer will be placed as the record treation, the weights values follow x's distribution and are different from each other if x's not a constant vector.  8. You have built a network using the tash activation for all the hidden units. You initialize the weights to relative large.
<ul> <li>Each neuron in the first hidden layer will perform the same computation. So even after multiple iterations of gradient decreat each neuron in the layer will be computing the same sting as other neurons.</li> <li>□ Each neuron in the first hidden layer will perform the anoughoutson in the first retain. But after one iteration of gradient decreant they will learn to compute different things because we have "to ken aymerter."</li> <li>○ 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 breasing", as described in lature.</li> <li>○ 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.</li> <li>✓ Correct</li> <li>1. Logistic regression's weights wishould be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to "break symmetry". True/Faile?</li> <li>○ True</li> <li>● Failse</li> <li>✓ Correct</li> <l< td=""></l<></ul>
gradient descent each neuron in the layer will perform the anex occupitation in the first indeed layer will perform the anex occupitation of gradient descent they will learn to compute different things because we have "broken symmetry".    Sach neuron in the first hidden layer will perform different things because we have "broken symmetry".   Sach neuron in the first hidden layer will compute the same thing, but neurons in different tayers 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.    Cerrett
of gradient descent they will learn to compute offerent things because we have "broken symmetry."  Bach neuron in the first hidden layer will compute same thing, but neurons in different layers will compute odifferent things, thus we have accomplished "symmetry treaking 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.  Cerrect  7. Logistic regression's weights w should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to "break symmetry", True/False?  True  False  Cerrect  Yes, Logistic Regression doesn't have a hidden layer, if you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer, so at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.
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.  Cerrect  7. Logistic regression's weights w should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fall to learn a useful decision boundary because it will fall to "break symmetry". True/False?  True  False  Correct  Yes, Logistic Regression doesn't have a hidden layer, if you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.  8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large  1/1 point
7. Logistic regression's weights w should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to 'break symmetry', True/False?  True  False  Correct  Yes, Logistic Regression doesn't have a hidden layer, if you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second literation, the weights values follow x's distribution and are different from each other if x is not a constant vector.  8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large
<ul> <li>7. Logistic regression's weights w should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to "break symmetry", True/False?</li> <li>○ True</li> <li>ⓒ False</li> <li>✓ Correct  Yes. Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.</li> <li>8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large</li> </ul>
then logistic regression will fail to learn a useful decision boundary because it will fail to "break symmetry", True/False?  True  False  Correct  Yes, Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.
<ul> <li>✓ Correct         Yes, Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.     </li> <li>8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large</li> </ul>
<ul> <li>Correct         Yes, Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.     </li> <li>You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large</li> </ul>
Yes, Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.  8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large
distribution and are different from each other if x is not a constant vector.  8. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large
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.
O 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.  O This will cause the inputs of the tanh to also be very large, thus causing gradients to also become large. You
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 be very small 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.
Correct Yes. tanh becomes flat for large values, this leads its gradient to be close to zero. This slows down the optimization algorithm.
Consider the following A bidder becomes to the
9. Consider the following 1 hidden layer neural network:
$x_1 - a_2^{(1)}$
$x_2$ $a_1^{[1]}$ $\hat{y}$
$a_4^{[1]}$
Which of the following statements are True? (Check all that apply).
$lacksquare b^{[1]}$ will have shape (4, 1)
✓ Correct
$ ightharpoonup W^{[1]}$ will have shape (4, 2)
✓ Correct
$oxed{\ }$ $b^{[1]}$ will have shape (2, 1)
$ ullet W^{[2]}$ will have shape (1, 4)
$igwedge W^{[2]}$ will have shape (1, 4) $igwedge Correct$
$igwedge W^{[2]}$ will have shape (1, 4) $igwedge C$ Correct $igwedge B^{[2]}$ will have shape (4, 1)
$\checkmark$ Correct $b^{[2]}$ will have shape (4, 1) $W^{[2]}$ will have shape (4, 1)
$\checkmark$ Correct $b^{[2]}$ will have shape (4, 1)
$\checkmark$ Correct $b^{[2]}$ will have shape (4, 1) $W^{[2]}$ will have shape (4, 1)
$\checkmark$ <b>correct</b> $b^{[2]}$ will have shape (4, 1) $W^{[2]}$ will have shape (4, 1) $b^{[2]}$ will have shape (1, 1) $\checkmark$ <b>correct</b> 10. In the same network as the previous question, what are the dimensions of $Z^{[1]}$ and $A^{[1]}$ ?
✓ correct $ b^{[2]} \text{ will have shape (4, 1)} $ $ W^{[2]} \text{ will have shape (4, 1)} $ $ b^{[3]} \text{ will have shape (1, 1)} $ ✓ correct

Correct