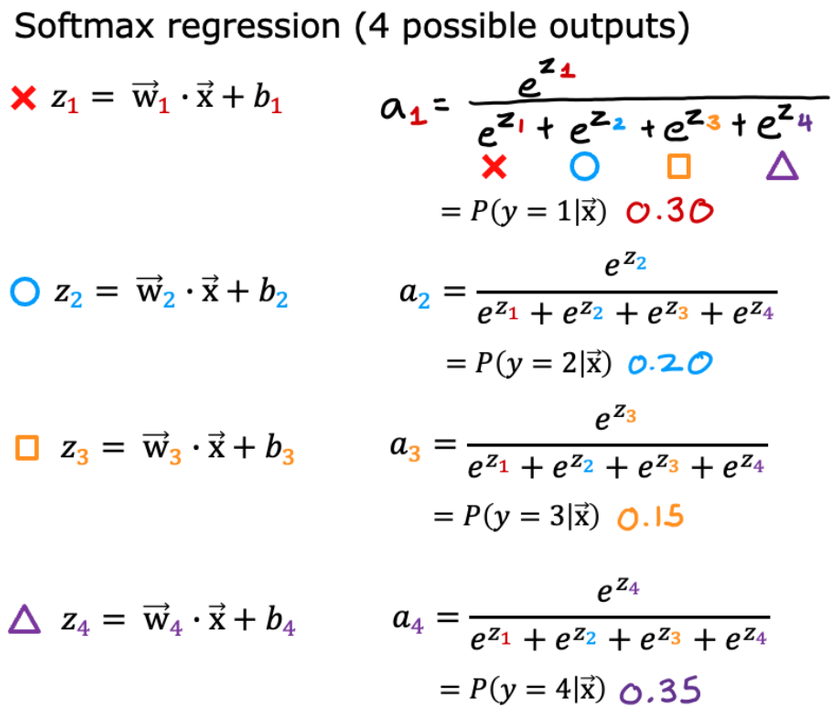
1. 1 / 1 point



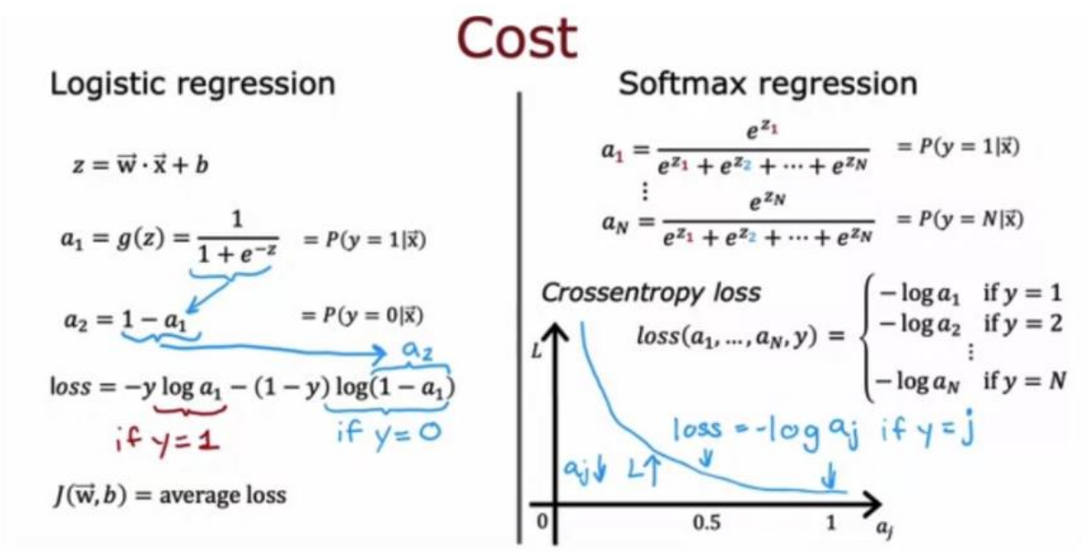
For a multiclass classification task that has 4 possible outputs, the sum of all the activations adds up to 1. For a multiclass classification task that has 3 possible outputs, the sum of all the activations should add up to ….

- (●) 1
- ( ) Less than 1
- ( ) More than 1
- ( ) It will vary, depending on the input x.

Correct

Yes! The sum of all the softmax activations should add up to 1. One way to see this is that if $e^{z_1} = 10, e^{z_2} = 20, e^{z_3} = 30$, then the sum of $a_1 + a_2 + a_3$ is equal to $\frac{e^{z_1}+e^{z_2}+e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}}$ which is 1.

2. 1 / 1 point



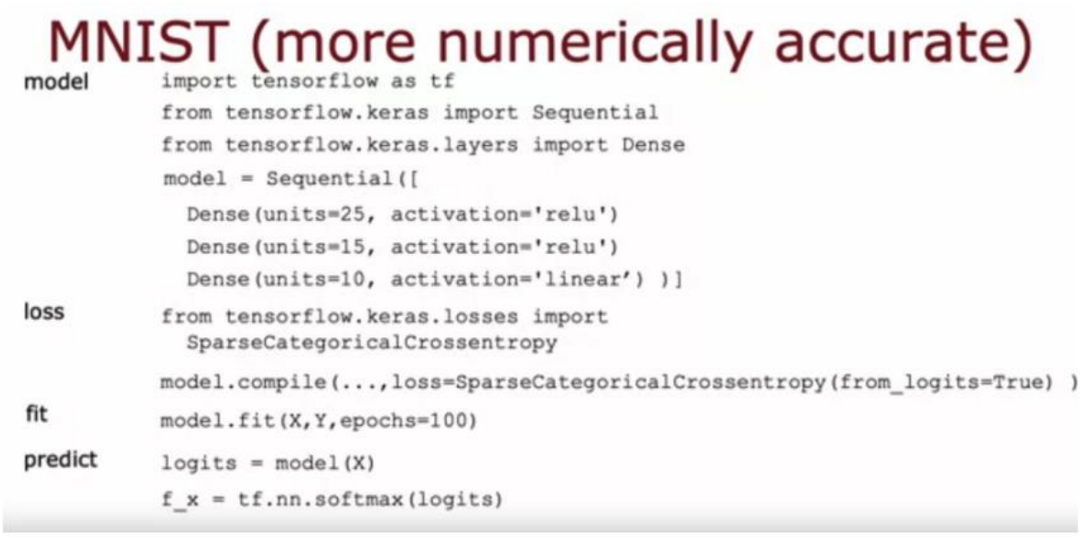
For multiclass classification, the cross entropy loss is used for training the model. If there are 4 possible classes for the output, and for a particular training example, the true class of the example is class 3 (y=3), then what does the cross entropy loss simplify to? [Hint: This loss should get smaller when a_3 gets larger.]

- (●) $-log(a_3)$
- ( ) z_3
- ( ) $\frac{-log(a_1)+-log(a_2)+-log(a_3)+-log(a_4)}{4}$

- z_3/(z_1+z_2+z_3+z_4)

Correct
Correct. When the true label is 3, then the cross entropy loss for that training example is just the negative of the log of the activation for the third neuron of the softmax. All other terms of the cross entropy loss equation ($-log(a_1), -log(a_2), and -log(a_4)$) are ignored

3. **1 / 1 point**



For multiclass classification, the recommended way to implement softmax regression is to set from_logits=True in the loss function, and also to define the model's output layer with…

- a 'softmax' activation
- a 'linear' activation

Correct
Yes! Set the output as linear, because the loss function handles the calculation of the softmax with a more numerically stable method.