EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the Privacy Policy.





MITx: 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

<u>Help</u>



<u>sandipan_dey</u>

<u>Course</u> > <u>Unit 3 Neural networks (2.5 weeks)</u> > <u>Project 3: Digit recognition (Part 2)</u> > 3. Activation Functions

3. Activation Functions

The first step is to design the activation function for each neuron. In this problem, we will initialize the network weights to 1, use **ReLU** for the activation function of the hidden layers, and use an identity function for the output neuron. The hidden layer has a bias but the output layer does not. Complete the helper functions in neural_networks.py, including rectified_linear_unit and rectified_linear_unit_derivative, for you to use in the NeuralNetwork class, and implement them below.

You will be working in the file part2-nn/neural_nets.py in this problem

Rectified Linear Unit

2/2 points (graded)

First implement the ReLu activation function, which computes the ReLu of a scalar.

Note: Your function does not need to handle a vectorized input

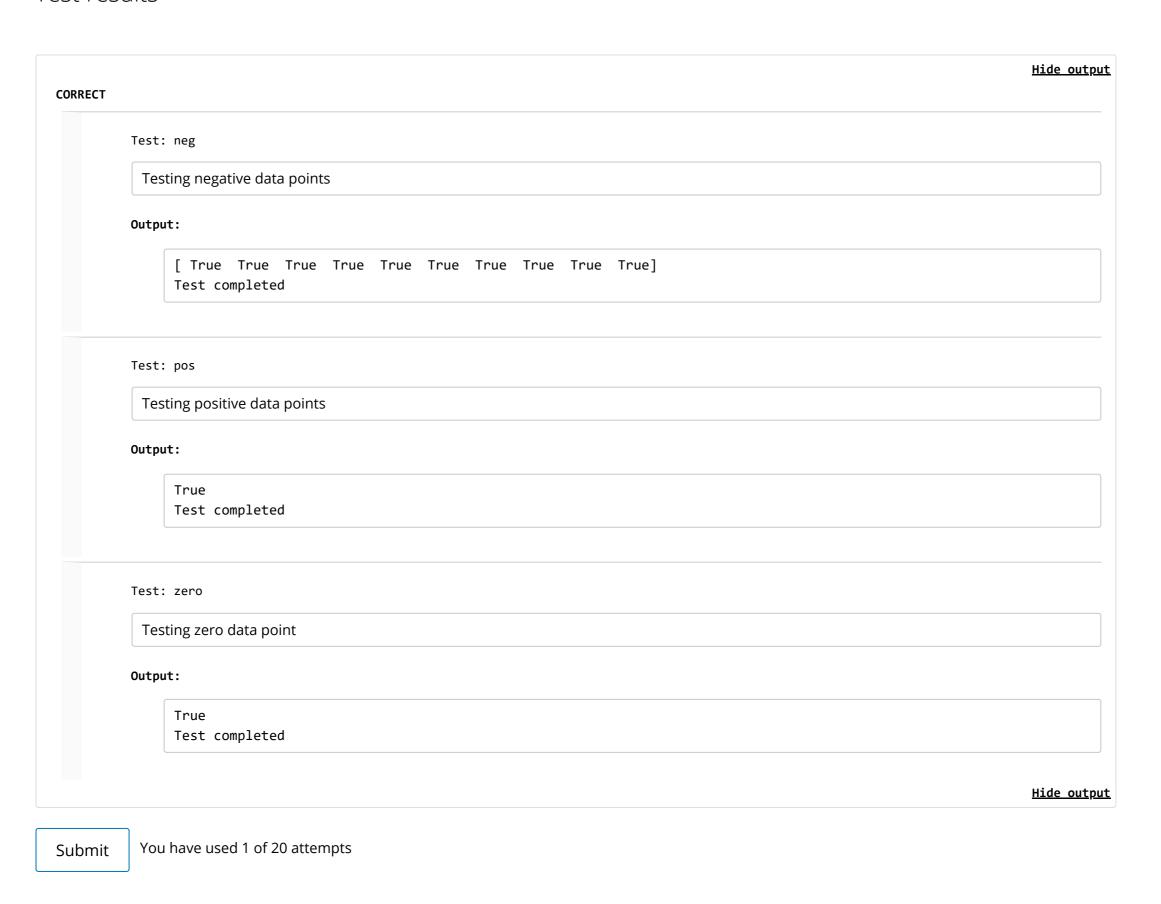
Available Functions: You have access to the NumPy python library as np

```
1 def rectified_linear_unit(x):
     """ Returns the ReLU of x, or the maximum between 0 and x."""
     return np.maximum(0, x)
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results



https://courses.edx.org/courses/course-v1:MITx+6.86x+1T2019/courseware/unit_3/P3_neuralnets/

✓ Correct (2/2 points)

Taking the Derivative

2/2 points (graded)

Now implement its derivative so that we can properly run backpropagation when training the net. Note: we will consider the derivative at zero to have the same value as the derivative at all negative points.

Note: Your function does not need to handle a vectorized input

Available Functions: You have access to the NumPy python library as np

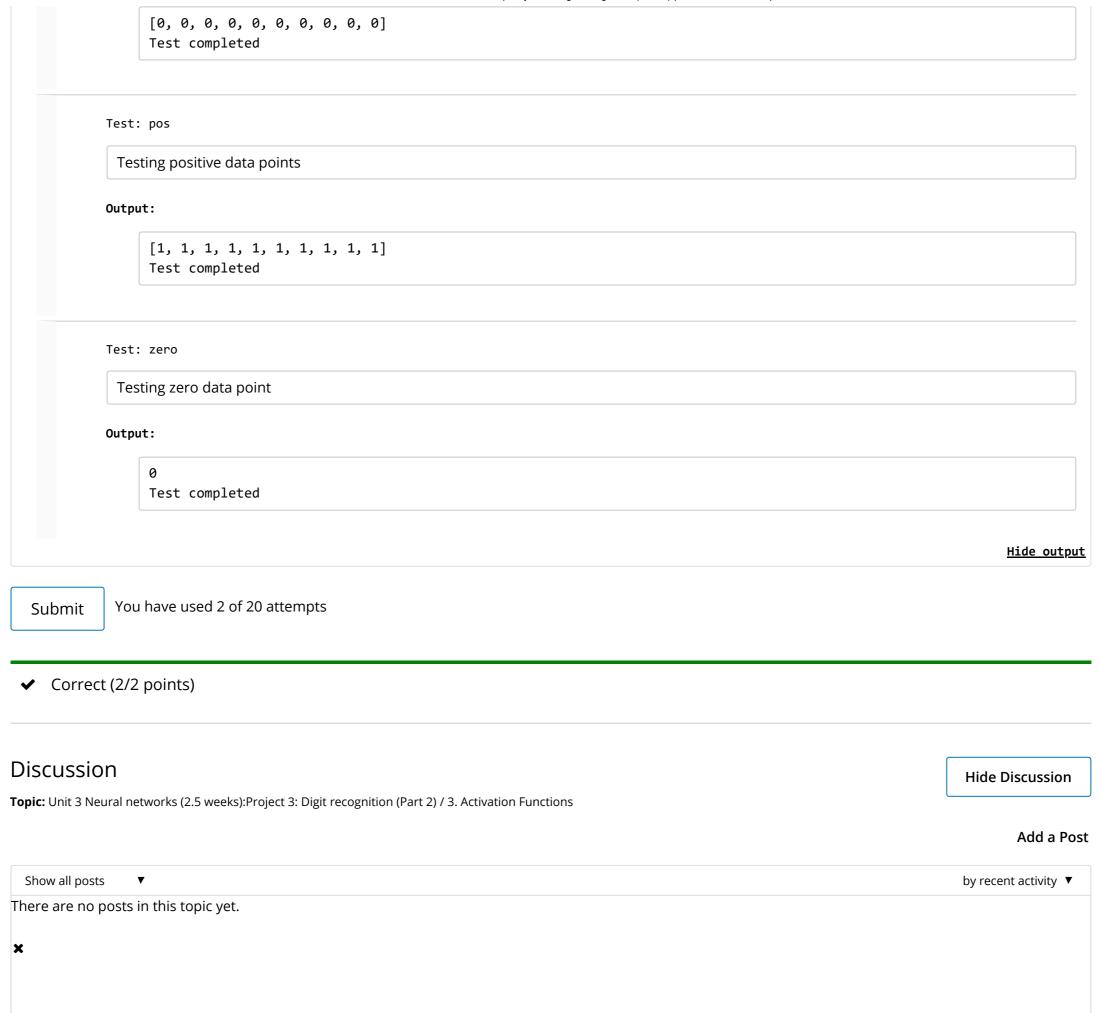
```
1 def rectified_linear_unit_derivative(x):
2    """ Returns the derivative of ReLU."""
3    return 1 if x > 0 else 0
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results





Learn About Verified Certificates

© All Rights Reserved