

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

Audit Access Expires May 11, 2020

You lose all access to this course, including your progress, on May 11, 2020.

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

2.0/2.0 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

```
def rectified_linear_unit(x):
    """ Returns the ReLU of x, or the maximum between 0 and x."""
    if x <= 0:
        return 0</pre>
```

```
5 else:
6 return x
```

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

Correct

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

Test results

See full output

CORRECT

See full output

Submit

You have used 3 of 25 attempts

1 Answers are displayed within the problem

Taking the Derivative

2.0/2.0 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    if x <=0:
4       return 0
5    else: return 1</pre>
```

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

Correct

```
def rectified_linear_unit_derivative(x):
    """ Returns the derivative of ReLU."""
    if x <= 0:
        return 0
    else:
        return 1</pre>
```

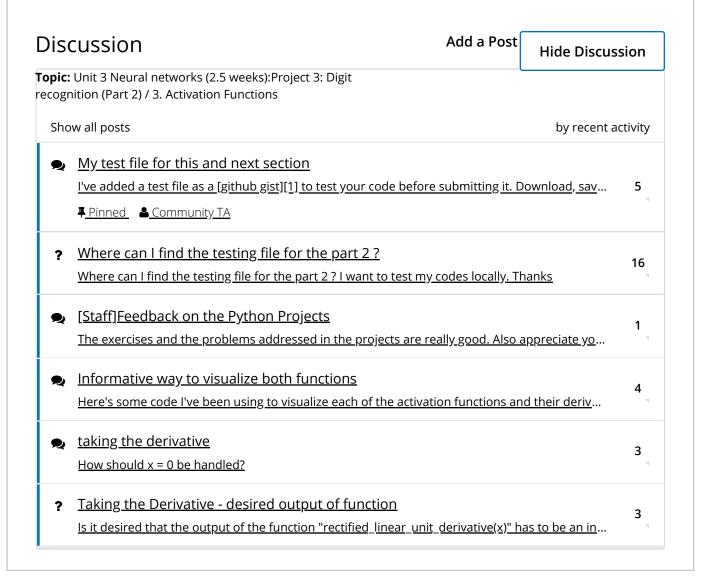
Test results

CORRECT
See full output
See full output

Submit

You have used 1 of 25 attempts

1 Answers are displayed within the problem



© All Rights Reserved