



[Course](#) > [Unit 3 Neural networks \(2.5 weeks\)](#) > [Project 3: Digit recognition \(Part 2\)](#) >
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
```

```
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

CORRECT

[See full output](#)

[See full output](#)

Submit

You have used 3 of 25 attempts

i 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
```

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
```

Test results

CORRECT

[See full output](#)

[See full output](#)

Submit

You have used 1 of 25 attempts

i Answers are displayed within the problem

Discussion









Add a Post

Hide Discussion

Topic: Unit 3 Neural networks (2.5 weeks):Project 3: Digit recognition (Part 2) / 3. Activation Functions

Show all posts

by recent activity

	<u>My test file for this and next section</u> I've added a test file as a [github gist][1] to test your code before submitting it. Download, sav...  Pinned  Community TA	5
	<u>Where can I find the testing file for the part 2 ?</u> Where can I find the testing file for the part 2 ? I want to test my codes locally. Thanks	16
	<u>[Staff]Feedback on the Python Projects</u> The exercises and the problems addressed in the projects are really good. Also appreciate yo...	1
	<u>Informative way to visualize both functions</u> Here's some code I've been using to visualize each of the activation functions and their deriv...	4
	<u>taking the derivative</u> How should $x = 0$ be handled?	3
	<u>Taking the Derivative - desired output of function</u> Is it desired that the output of the function "rectified_linear_unit_derivative(x)" has to be an in...	3