MITx: 6.86x
Machine Learning with Python-From Linear Models to Deep Learning

Help sandipan_dey

Course > Unit 3 Neural networks (2.5 weeks) > Project 3: Digit recognition (Part 2) > 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

```
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

Hide output

CORRECT

Test: neg

Testing negative data points

Output:

```
[ True  True  True  True  True  True  True  True  True  True]
Test completed
```

Test: pos

Testing positive data points

Output:

```
True
Test completed
```

Test: zero

Testing zero data point

Output:

```
True
Test completed
```

Hide output

Submit

You have used 1 of 20 attempts

✔ 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

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

```

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

Correct

Test results

Hide output

CORRECT

Test: neg

Testing negative data points

Output:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Test completed
```

Test: pos

Testing positive data points

Output:

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Test completed
```

Test: zero

Testing zero data point

Output:

```
0
Test completed
```

Hide output

Submit

You have used 2 of 20 attempts

✓ Correct (2/2 points)

Discussion

Hide Discussion

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

Add a Post

Show all posts ▼ by recent activity ▼

There are no posts in this topic yet.

✖