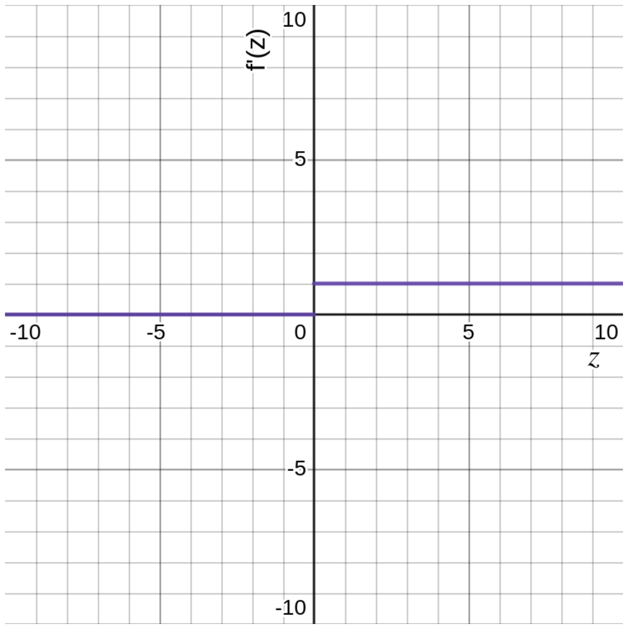
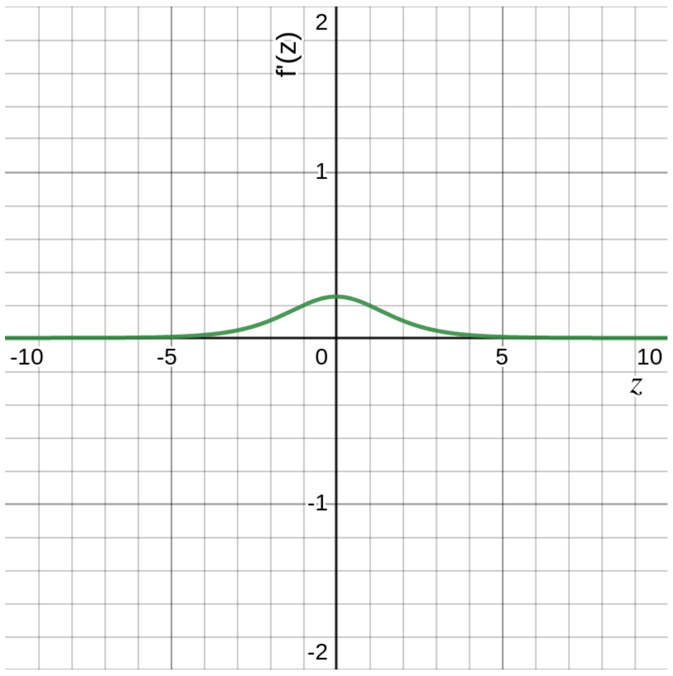
Problem 1 (20 points). As neural networks are typically trained using (stochastic) gradient descent optimization algorithms, properties of the activation functions affect the learning. Here we divide the domain of an activation function into:

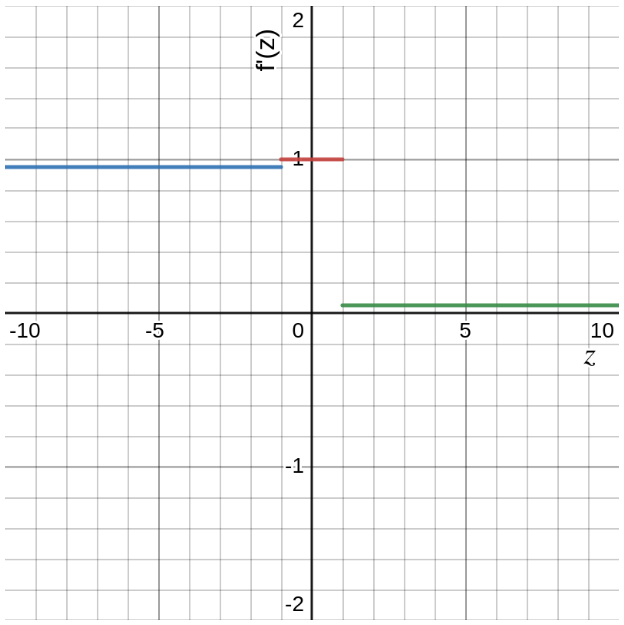
1. **fast learning region**, if the magnitude of the gradient is larger than 0.9,



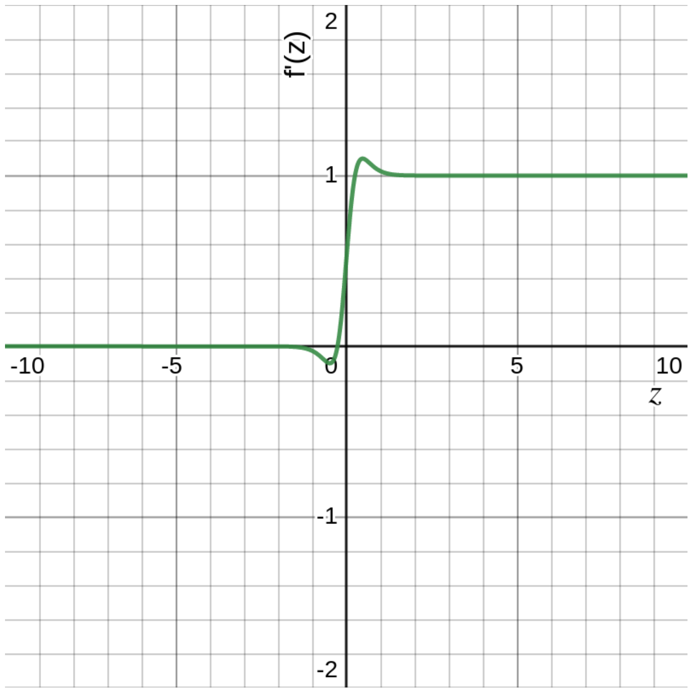
2. **active learning region**, if the magnitude of the gradient is between 0.1 and 0.9 (inclusive)



3. **slow learning region**, if the magnitude of the gradient is larger than 0 but smaller than 0.1, and



4. **inactive learning region**, if the magnitude of the gradient is 0.



Problem 2 (15 points). Answer the following questions regarding the back propagation algorithm.

1. **(10 points).** Describe in your words Algorithm 6.1 and Algorithm 6.2 in the Deep textbook (on pages 202 and 204). For each algorithm, you need to describe what each line is doing and why the algorithm would work correctly.

Algorithm 6.1 is as follows:

```
For each i in (1…N), create a new graph node from Xi
For each i in (head node+1… N):
    Set Ai is the arguments from the previous nodes
    Apply the function f on the set Ai, assign to ui
Return final node, un
```

Algorithm 6.2 is as follows:

```
Gradient table is initialized to store computed gradients.
For N…1, compute gradient using sum of the gradient for previous vals and store it in
gradient table
Return gradient table
```

2. **(5 points).** Explain in your own words why the two algorithms require “the same order of the number of computations”. (See page 204 of the textbook for more information.)

I did not find this quote in the textbook but I can take a crack at it. I assume that because algorithm 1 actually “loops” the data twice, it evens out the amount of time that algorithm 2 takes doing calculations.

Problem 3 (20 points). Using the framework you have established, design and train a neural network to learn how to approximate

$$g(x) = 1 + sin\,(3\pi x/2)$$

Note that your neural network should have as few as possible neurons in the hidden layer(s) and the largest error (i.e., the absolute difference between g(x) and the output of your neural network) should be no more than 0.05 in the range of inputs from -2 to 2. You can generate and use no more than 100 training samples.

Submitted as function_approx.py

Extra Credit. Problem 4 (20 points). The weights and biases of a softmax layer (implemented using equations (6.28) and (6.29) in the Deep Learning textbook) are available for download from Canvas as hw2_softmax_weights.m Download hw2_softmax_weights.m . (Note that they are given as a MATLAB program and you should be able to extract the numbers easily from the file.)

1. **(2 points).** What is the architecture of the softmax layer? In other words, how many inputs, how many neurons, and how many connections are there in the layer?

The softmax layer has 20 bias values associated with it. Since there are also 2000 weights associated, there are a total of 100 neurons with 100 connections per layer.

2. **(4 points).** Classify the following example, which can be downloaded from Canvas as hw2_softmax_sample.txt ☐ Download hw2_softmax_sample.txt .

In this case, the softmax layer has determined that it thinks that there are 10 possibilities out of the entire set of choices it could make. It leans most heavily towards whatever it has weighed as 1.606391429901123.

3. **(8 points).** Suppose that we use cross entropy loss, the learning rate is 0.1, and the correct label for the sample is the first class (i.e., class 1 if you labels the classes as 0, 1, 2, …, 19), after we apply one step gradient decent optimization using the given sample in (2), how many of the weights and biases will be increased, decreased, or remain the same respectively? For numerical stability, we consider that any change smaller than 0.001 is the same.
4. **(6 points).** Compute a small vector so that the new sample created by adding the vector to the given sample will be classified as from the second class. The length of the vector (in L_2 norm) should be as small as possible. Explain how you have obtained your solution and confirm that the resulting sample is indeed classified as from the second class.