4. Training the Network

Project due Apr 5, 2023 08:59 -03 Completed

Forward propagation is simply the summation of the previous layer's output multiplied b
wire, while back-propagation works by computing the partial derivatives of the cost fur
every weight or bias in the network. In back propagation, the network gets better at mi
predicting the output of the data being used for training by incrementally updating thei
using stochastic gradient descent.

We are trying to estimate a continuous-valued function, thus we will use squared loss a
an identity function as the output activation function. $f(x)$ is the activation function th
to our final layer output node, and $\hat{a}$ is the predicted value, while y is the actual value o

$$C = \frac{1}{2}(y - \hat{a})^2$$

$$f(x) = x$$

When you're done implementing the function `train` (below and in your local repository
if the errors are decreasing. If your errors are all under 0.15 after the last training iterati
implemented the neural network training correctly.

You'll notice that the `train` function inherits from `NeuralNetworkBase` in the codebox
grading purposes. In your local code, you implement the function directly in your Neura
one file. The rest of the code in `NeuralNetworkBase` is the same as in the original `Neu`
have locally.

**In this problem, you will see the network weights are initialized to 1. This is a bad sett
do so for simplicity and grading here.**

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

Implementing Train

5.0/5.0 points (graded)

Available Functions: You have access to the NumPy python library as `np`, `rectified`
`output_layer_activation`, `rectified_linear_unit_derivative`, and
`output_layer_activation_derivative`

Note: You need to use `output_layer_activation` at least once for the grader to corre

```
15        hidden_layer_activation = relu_vec(hidden_layer_weighted_input)
```

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

Unanswered

Previous Next

Submit You have used 7 of 50 attempts