

Gradient Descent and Error Backpropagation

Question 2: Implementing Backpropagation

By this time, you should have done the Exercises for week 3. If not, go and do them now. ... I'll wait.

Download the notebook `as03.ipynb` and the module `utils.py`. The notebook imports and uses the `utils` module, and also has a number of class definitions that you will use and modify. Note that you will be using the `Logistic` and `CrossEntropy` classes that you (hopefully) completed in the previous assignment. Their full implementations will be available after 11:00pm on Monday.

In this question, you will complete the implementation of learning by error backpropagation. In doing so, you will complete the functions `backprop` and `learn` in the `Network` class. If you've done it correctly, you can use your code to learn to classify the `UCclasses` dataset in the notebook. And that will make you feel warm inside.

Here are the specific tasks:

- (a) `backprop`: **Complete the `Network.backprop` function**, which performs an update to the network weights and biases using the error backpropagation algorithm. The method uses the **current (saved) network state**, including the activities of the output layer, and compares them to the targets.

You should assume that all layers after the input layer are of the `DenseLayer` class.

You should use the `derivative` methods of the loss function and activation functions.

- (b) `learn`: **Complete the `Network.learn` function**, which tries to find the optimal network weights and biases. This function should call `backprop` to update the weights and biases.
- (c) `accuracy`: **Complete the function `accuracy` at the end of the notebook**. This function takes the output from a network, as well as the targets, and calculates the fraction of correctly classified samples.

There is some code at the end of the notebook that creates a network, adds some layers, and then calls the `learn` function to train it on the dataset. Your implementation should be able to achieve over 98% accuracy in about 5000 epochs.

Submit the updated notebook to Kritik with the filename `as03.ipynb`. Remember to ensure that nothing in the file can be used to identify you; the peer-assessment process is supposed to be anonymous.