CSC 737: Assignment 2

Total: 90 points

Problems

(90 points) In this programming assignment you would design the back-propagation algorithm for a three layer network (input, hidden and output). Each layer contains activated neurons where the activation function at the input layer is identity, i.e., $a^{(1)} = x^{(1)}$. For other layers (hidden and output), the user can have a choice of identity (or linear i.e., $g(z) = z$), sigmoid or ReLU activation function at its neurons. The number of neurons per layer, the choice of loss function and the activation function are provided by the user as design choices i.e., these are provided as input for the class constructor defining your network.
Implementation (40 points)

1. You need to implement your network as a class that stores the number of neurons in each layer, the weights, the bias, the corresponding loss derivative with respect to each weights and the activation function used in each layer. It might be useful to cache the error terms for re-use. The constructor takes in the number of input and output connections in the hidden layer and the output layer and the corresponding activation function to use. The number of output connections from the hidden layer should match the number of input connections in the output layer.

2. You need to implement the forward and backward pass for the network. The forward pass will take the input features of a given batch size and calculate the final output ($\hat{y}$) using forward propagation. The backward pass will take gradient tensors of the output layer as input of a given batch size and calculate the gradients of all the layers and subsequently updates the weights at each layer.

3. You need to implement for each type of loss function: cross-entropy and mean squared error, given predicted output $\hat{y}$ and actual output y as arguments, what are the gradients of the loss and the scalar loss value.

4. It might be useful to create separate functions to compute the activation value and derivative values for each different activation function and call the respective functions in weight updates based on user preference.

5. Do not use autograd function. You can use autograd to validate your results. It is recommended to use Pytorch for your programming library. If you are using other libraries please mention the same in the report.

Report (50 points) Your report should include the implementation details stating what functions you implemented in the code and the parameters you are using. You need to evaluate your implementation for the following experimental objectives:

1. **How the different activation functions affect the weight updates in mean squared loss function (regression task)?** Do not use identity function in the hidden layer neurons.

2. **How the different activation functions affect the weight updates in cross entropy loss function (multi-label classification task)?**

For each of the experiments,

1. Generate random x and y values to create your training instances and validation instances.

2. State the number of instances you are creating, how you are creating them, are you standardizing or normalizing your input features, how are you initializing your weights (ranges and distribution) and the hyper-parameter values you are using in the experimental details.

3. You need to plot the both the training error and validation error over time for each experiments and state your inferences based on the plots.

General Instructions

1. Submit a single PDF or Word file for your report and paper review.

2. Submission of code need to be in code submission and report in report submission for assignment. Any cross submission will result in penalty.

3. Codes need to be implemented in Python and submitted as a single zip file. Code needs to be modularized and properly documented with comments.

4. Variable naming in the code need to be descriptive or corresponding to the definition in the report.

5. Write the code and report by yourself. Do not copy code or report.

6. Do not fabricate results. If you submit results in report without software implementation to support it, your results is not corroborated and will be considered fabrications.

7. Type in your answers mostly. Do not hand-write, scan and upload.

8. All problems need to be indexed as per their indexing in the assignment.

9. Provide list of references which include books, web-sites, people and papers.

10. No late assignment accepted.