Assignment 3 - Multi Layer Perceptron

Advanced Topics in Neural Networks

17 October 2023

Homework

Implement a Multi Layer Perceptron (MLP) using raw PyTorch Tensor operations. Avoid using APIs such as **torch.nn** or **torch.functional**.

The MLP architecture should consist of 784 input neurons, 100 hidden neurons, and 10 output neurons. The tasks for this assignment and their respective points are as follows:

1. Implement forward propagation. (1 point)
2. Implement the backpropagation algorithm using the chain rule. Refer to the provided resources. (6 points, -1 point if the update is not correct)
3. Utilize batched operations for enhanced efficiency. (2 point)
4. Use the MNIST dataset to evaluate your implementation. Measure the loss and the accuracy for both training and validation. For loss calculation, you are allowed to use torch.nn.functional.cross_entropy(). (-3 points if not done)
5. Aim for achieving 95% accuracy on the validation split of the MNIST dataset. Ensure reproducibility on Google Colab, in under 5 minutes of runtime. (1 point)
    - Things you may try to improve the accuracy: changing the learning rate, activation functions, data augmentation.
    - More advanced things you can try (but you have to research them ahead time): Dropout, Batch Normalization, Learning Rate Decay, Weight Initialization, Regularization, Weight Decay, Gradient Clipping.
    - Attention, you are allowed to use only raw tensor operations, without using Data Augmentation libraries or high level APIs like torch.nn and torch.functional.

Try to run your model both on CPU and GPU and check the runtime. If you do not have a GPU available, use Google Colab and copy your script in a notebook cell (your script should be able to run on both CPU and GPU depending on a device parameter passed from the outside).

Bonus: Strive for higher accuracy (might not be graded, but always try to be better).