A.I. Assignment 7

PyTorch

Task 1.

Install the pytorch module. For instructions look at the following link: pytorch.org.

Task 2.

Following the example included in the folder example create a neural network (feed forward, fully connected, linear activation function in all nodes) with one hidden layer (or more if you want to) that will give a good approximation of the function

$$f: [-10, 10]^2 \to R,\ f(x_1, x_2) = sin(x_1 + \frac{x_2}{\pi}).$$

Suggested steps for the workflow:

FIRST STEP

in the file *createdb.py*:

1. Create the training database:
    a. Create a distribution of 1000 random points in the domain $[-10, 10]^2$
    b. Compute the value of the function *f* for each point
    c. Create the pairs $d^i = ((x_1^i, x_2^i), f(x_1^i, x_2^i)),\ i = \overline{1, 1000}$

    Helping functions: *torch.sin, torch.cos, torch.add, torch.rand, torch.full_like, torch.addcmul, torch.column_stack, …*

2. Save the database into the file *mydataset.dat*

    Help function: *torch.save*

SECOND STEP

A. In the file *myModel.py,* declare a class Net that suits your problem (the one from example should work if you require just one hidden layer with linear activation).
B. in the file *trainmodel.py*:
    1. Load the training data from the file *mydataset.txt*

        Help function: *torch.load*

    2. Following the example from file *train_Batch.py* declare and train your ANN
    3. Save your trained network in file *myNetwork.pt* (there are several ways to save/load a model, look at the example for the most used one)

THIRD STEP

In the file *inference.py* write a small program that loads and uses the trained network in order to approximate the function values for the user's input values.

Advise before developing your application:

- make a graphic with the loss versus epochs to see how it changes over training, observe how the variation of the parameters affect its pattern
- play with the parameters on the example, see how the performance of the train model varies with the error, increase/decrease the learning rate, batch sizes, and the number of epochs
- play with the architecture of the model - increase/decrease the number of neurons from the hidden layer or add a new layer or erase the ReLU function applied to the hidden layer

For this assignment one can get a maximum **100** points.

Due time: 1 working week for the final solution.

IF is not done in the first week you will have a penalty of 10 points.

The solution can not be turned in after 2 working weeks.