Assignment #1: due week 5 just before class at 1830 hours

Question #1

Given a fully connected Neural Network as follows:

a. Input (x1,x2,…,xd): d-nodes
b. K-hidden fully connected layers with bias of 2d+1 nodes
c. Output (predict): 1 node
d. Use Relu activation function for all layers

1. Implement this neural network in pytorch

2. Generate the input data (x1,x2,..xd) \in [0,1] drawn from a uniform random distribution

3. Generate the labels y = (x1*x1+x2*x2+…+xd*xd)/d

4. Implement a loss function L = (predict-y)^2

5. Use batch size of 1, that means feed data one point at a time into network and compute the loss. Do one time forward propagation with one data point.

6. Compute the gradients using pytorch autograd:
    a. dL/dw, dL/db
    b. Print these values into a text file: torch_autograd.dat
7. Implement the forward propagation and backpropagation algorithm from scratch, without using pytorch autograd, compute the gradients using your implementation
    a. dL/dw, dL/db
    b. Print these values into a text file: my_autograd.dat
8. Compare the two files torch_autograd.dat and my_autograd.dat and show that they give the same values up to 5 significant numbers
9. Use K=10,d=10

Question #2

Run the following code, generate the computational graph, label and explain **all** nodes (all nodes means not just the leave nodes, all intermediate nodes should be explained):

```
import torch
import torch.nn as nn
from torchviz import make_dot

# ====================================================
```

```
def print_compute_tree(name,node):
    dot = make_dot(node)
    #print(dot)
    dot.render(name)
# ====================================================

if __name__=='__main__':

    torch.manual_seed(2317)
    x = torch.randn([1,1,10],requires_grad=True)
    cn1 = nn.Conv1d(1,1,3,padding=1)
    fc1 = nn.Linear(10,10)
    fc2 = nn.Linear(10,1)

    y = torch.sum(x)

    c = cn1(x)
    x = torch.flatten(x)+torch.flatten(c)
    x = fc1(x)
    x = fc2(x)

    loss = torch.sum((x-y)*(x-y))

    print_compute_tree('./tree_ex' ,loss)


 ######################################################
```

Submission:
Submit your code and report (max 2 pages for each question):

1. One copy in NTULearn.
2. One copy in your course GitHub project