Homework 2

Neural Networks - Feedforward Pass

Xiaoyu Xiang(xiang43@purdue.edu)

Platform and Packages

- Python 3.6, PyTorch 0.4.1
- Numpy

Code Implementation

neural_network.py

In this script, we define a class NeuralNetwork, to use this class:

```
from neural_network import NeuralNetwork
```

It takes 1 input, that defines the size of input for each layer.

`NeuralNetwork.getlayer(layer_index: int)` can assign the weights for each layer. If not assigned, each layer will take random numbers(mean = 0, std = 1/sqrt(layer size)) as weights.

`NeuralNetwork.forward(input: DoubleTensor)` can do the forward pass. For each layer, the output `y` has the following relationship with the input `x`:

`y = Wx + b`

Where the b is the bias node.

logic_gates.py

This script defines 4 classes: `AND`, `OR`, `NOT`, `XOR`.

All the classes take bool input and give bool outputs.

For `AND`, `OR`, `NOT`, they are generated by 1-layer neural network. By transferring the input bool to int, we can apply the linear formula to calculate the output. Where the weights for each logic gates are assigned as below:

Gate	b	w1	w2
AND	-10	7	7
OR	-1	7	7
NOT	5	-7	\

For `XOR`, we cannot express it with one layer neural network that satisfies all the equations. So we generate a 2-layer network with the following parameters:

Layer	b	w1	w2
0	[-5, -5]	[6, -6]	[-6, -6]

Layer	b	w1	w2
1	-5	7	7

Test Result

Run `'python test.py'`

For each logic gate, the test results are listed in the tables below.

AND

Input 1	Input2	Output
True	True	True
True	False	False
False	True	False
False	False	False

OR

Input 1	Input2	Output
True	True	True
True	False	True
False	True	True
False	False	False

NOT

Input	Output
True	False
False	True

XOR

Input 1	Input2	Output
True	True	False
True	False	True
False	True	True
False	False	False