```python
from numpy import abs
from numpy import exp
from random import random


def un_normalized(value, min, max):
    return value * (max - min) + min


class Neuron:

    def __init__(self, weight):
        self.weight = weight
        self.output = list()
        self.delta = list()


class BackpropagationNN:

    def __init__(self, input_layer, hidden_layer, output_layer, learning_rate):
        self.INPUT_LAYER = input_layer
        self.HIDDEN_LAYER = hidden_layer
        self.OUTPUT_LAYER = output_layer
        self.LEARNING_RATE = learning_rate

    def __str__(self):
        str_buff = ""
        str_buff += "Input to Hidden\n"
        for neuron in self.net[0]:
            str_buff += str(neuron.weight) + "\n"
        str_buff += "Hidden to Output\n"
        for neuron in self.net[1]:
            str_buff += str(neuron.weight) + "\n"
        return str_buff
```